Hosted Payment Page

Implementation Guide

Document version 3.37.1

SOCIETE GENERALE

DEVELOPPONS ENSEMBLE
L'ESPRIT D'EQUIPE

Contents

1. HISTORY OF THE DOCUMENT

Version	Author	Date	Comment
3.37.1	Société Générale	2/13/2024	• Update of the chapter *Defining the steps of the payment process - As seen by the buyer* Data dictionary: • **vads_threeds_mpi**
3.37	Société Générale	6/12/2023	Chapter(s) updated : • Offering additional payment attempts • Transmitting sub-merchant details Data dictionary: • **vads_payment_cards** • **vads_operation_type** • **vads_risk_assessment_result** • **vads_threeds_auth_type** • **vads_threeds_mpi** • **vads_submerchant_company_type** • **vads_submerchant_company_name** • **vads_use_case**
3.36	Société Générale	12/15/2022	Chapter(s) updated : • Authorization request validity period Data dictionary: • **vads_currency** • **vads_acquirer_transient_data** • **vads_product_amountN** • **vads_cust_legal_name** • **vads_payment_cards**
3.35.9.1	Société Générale	11/29/2022	Update of the chapter i*Capture delay longer than the authorization validity period* Update of the chapter *Defining the steps of the payment process - As seen by the buyer*
3.35.9	Société Générale	11/14/2022	• Updated the signature calculation example for PHP code. Data dictionary: • Updated the values in the **vads_wallet.** field. • Updated the values in the **vads_threeds_auth_type.** field. • Update the timeout value for the **vads_redirect_success_timeout** and **vads_redirect_error_timeout** fields • Added the **vads_submerchant_state** and **vads_submerchant_facilitatorId.** fields • Update of the **vads_cust_legal_name.** field
3.35.8	Société Générale	8/17/2022	• Update of the list of compatible payment methods in several installments. Data dictionary:

Version	Author	Date	Comment
			• Updated the values in the **vads_risk_assessment_result.** field.
3.35.7	Société Générale	7/28/2022	Data dictionary: • Addition of the **vads_archival_reference** field. • Updated the **vads_acquirer_transient_data** field description.
3.35.6	Société Générale	5/25/2022	• Added the *Definitions* chapter. • Added schematic diagrams with the chaining reference in the chapters *Immediate payment*, *Deferred payment* and *Payment in installments*. • Added chapter *Offering payment in a foreign currency*. Data dictionary: • Updated the **vads_token_id** field description. • Updated the **vads_threeds_mpi** field description. • Updated the values in the **vads_card_nature** and **vads_card_product_category** fields. • Updated the values in the **vads_payment_error** field.
3.35.5	Société Générale	4/25/2022	• Updated the life cycle of payments in installments. Data dictionary: • Addition of the **vads_initial_issuer_transaction_identifier** field. • Added the fields **vads_card_nature** and **vads_card_product_category**. • Updated the CB network authorization return codes. • Updated the *Payment method details* chapter. • Updated the *Transaction details* chapter.
3.35.4	Société Générale	3/28/2022	• Added the **signature** field in the following chapters describing the parameters to be used per use case. Data dictionary: • Updated the values in the **vads_auth_result** and **vads_contracts** fields. • Updated the **vads_capture_delay** field description. • Updated the values in the **vads_payment_src** field. • Updated the **vads_payment_cards** field description. • Added the **vads_user_info** and **vads_pays_ip** field and in the *Buyer details* chapter.
3.35.3	Société Générale	1/26/2021	• Updated the chapter *Authorization request validity period*. • Updated the chapter *Processing Merchant ID*. Data dictionary: • Field value updates for **vads_acquirer_network**, **vads_auth_result**, **vads_contracts**, **vads_payment_cards**.
3.35.2	Société Générale	11/18/2021	• Updated the chapter *Payment in installments*. • Updated the chapter *Authorization request validity period*. Data dictionary: • Updated the format and values of the **vads_threeds_eci** field.

Version	Author	Date	Comment
			• Updated the format and values of the **vads_threeds_cavvAlgorithm** field. • Updated the **vads_sub_effect_date** field format. • Field value updates for **vads_acquirer_network**, **vads_contracts**, **vads_payment_cards**.
3.35.1	Société Générale	10/5/2021	• Updated the chapter *Defining the capture mode*. • Updated the chapter *Transmitting sub-merchant details*. Data dictionary: • Updated the **vads_auth_number** field format. • Updated the **vads_order_id** field description. • Updated the **vads_submerchant_name** field description. • Updated the **vads_validation_mode** field description.
3.35	Société Générale	9/1/2021	• Updated the chapter *Test and troubleshooting*. Data dictionary: • Updated currencies in **vads_currency**

2. DEFINITIONS

2.1. Authorization request

An authorization request is the operation that allows to accept or refuse a transaction.

It puts the cardholder's bank (SAE = Issuer Acceptance System) in contact with the merchant's bank (SAA = Acquirer Acceptance System) and the payment provider: the payment gateway.

When an authorization request is accepted, the authorization limit of the card is lowered by the authorized amount.

In the CB network, an accepted authorization request is valid:

- 7 days for Visa, Mastercard, Visa Electron, e-Carte Bleue and Vpay cards
- 30 days for Maestro cards

2.2. Information request

An information request is an operation that allows to verify the validity of the card, **without debiting it**.

This is a specific type of authorization request, the amount of which is 0€.

When the acquirer does not support information requests, the only way to verify a card is to make a EUR 1 authorization request, without capturing it at the bank.

Holders of prepaid and immediate debit cards will see a virtual debit of EUR 1 on their account.

Depending on the card type, the outstanding balance of the card is then restored when the issuer cancels the EUR 1 authorization request (up to 30 days for debit cards).

An information request is sent:

- For a deferred payment, if the capture date is beyond the authorization lifespan
- When creating a card token without a payment
- When updating a card token

Information requests (or EUR 1 authorizations, if applicable) are represented in the Merchant Back Office by a "**Verification**" type transaction.

2.3. Chaining of CIT/MIT transactions

The second Payment Services Directive (PSD2) introduced the need to authenticate the cardholder when initiating an e-commerce transaction.

It then becomes essential to identify whether the payment request is initiated:

- by the buyer:

  CIT (Customer Initiated Transaction): buyer-initiated transaction with buyer interaction.

E.g.: payment (or card registration) that requires card data entry or cardholder authentication.

- or by the merchant:

  MIT (Merchant Initiated Transaction): transaction initiated by the merchant, without the presence of the buyer, linked to an initial **CIT** transaction.

  E.g.: umpteenth installment of a payment in installments or of a recurrent payment.

A new principle appears for the following transaction authentications: **operation chaining**.

In the context of a **CIT** transaction, regulations require a cardholder authentication. In response to the request for authorization or information, the issuer returns a unique transaction identifier, hereafter referred to as the “chaining reference”. This chaining reference is then used in the **MIT** transactions to indicate to the issuer that the transaction is part of a series of payments, for which the cardholder authenticated him or herself in the first payment.

Without this information, the issuer can refuse an **MIT** transaction for lack of authentication (soft decline).

3. DIFFERENT TYPES OF PAYMENTS

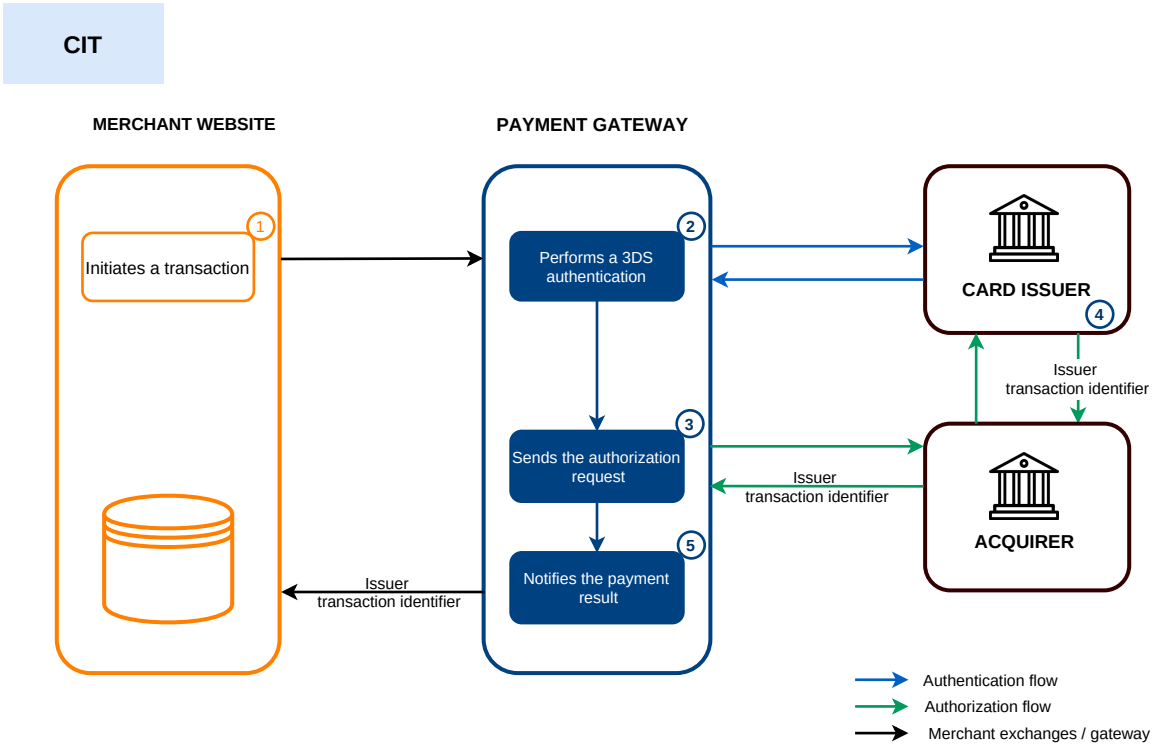
3.1. Immediate payment

A payment is considered as **immediate payment** if:

- the amount is debited once,
- the capture delay at the bank is 0 days.

The payment is captured at the bank as soon as possible.

Simplified diagram



1. The merchant site submits a payment request.
2. The payment gateway initiates the cardholder's authentication process with the issuer.

> The regulation imposes the cardholder authentication for all transactions ***CIT***.

3. Once the authentication (challenge or frictionless) is completed, the gateway proceeds with the authorization request by providing the cardholder's authentication details.
4. The issuer generates a unique transaction identifier and transmits it in the response to the authorization request.
5. The payment gateway notifies the merchant website about the payment result.

> The issuer transaction identifier is stored by the payment gateway at the transaction level.
>
> In case the merchant duplicates the transaction (***MIT***), the gateway automatically uses this identifier as a chaining reference.
>
> In this use case, the way the chaining reference is handled is transparent to the merchant.

3.2. Deferred payment

A payment is considered a **deferred payment** if:

- the amount is debited once,
- the capture delay is strictly more than 0 days.

  The capture date cannot be more than 12 months after the payment request registration date.

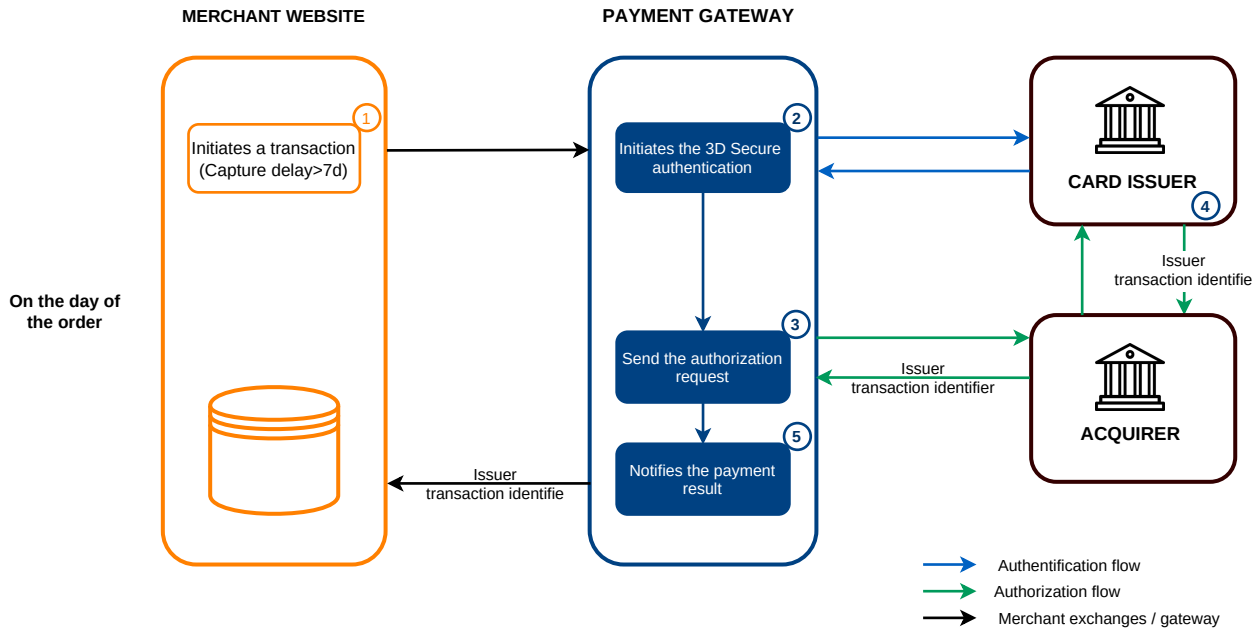
There are two types of deferred payments:

- **Capture delay shorter than the authorization validity period** (see: *Authorization request validity period* on page 22)
- **Capture delay longer than the authorization validity period** (see: *Authorization request validity period* on page 22)

3.2.1. Capture delay shorter than the authorization validity period

Simplified diagram

CIT



On the day of the order:

1. The merchant site submits a payment request.
2. The payment gateway initiates the cardholder's authentication process with the issuer.

   The regulation imposes cardholder authentication for all ***CIT*** transactions.

3. Once the authentication (challenge or frictionless) is completed, the gateway proceeds with the authorization request by providing the cardholder's authentication details.
4. The issuer generates a unique transaction identifier and transmits it in the response to the authorization request.

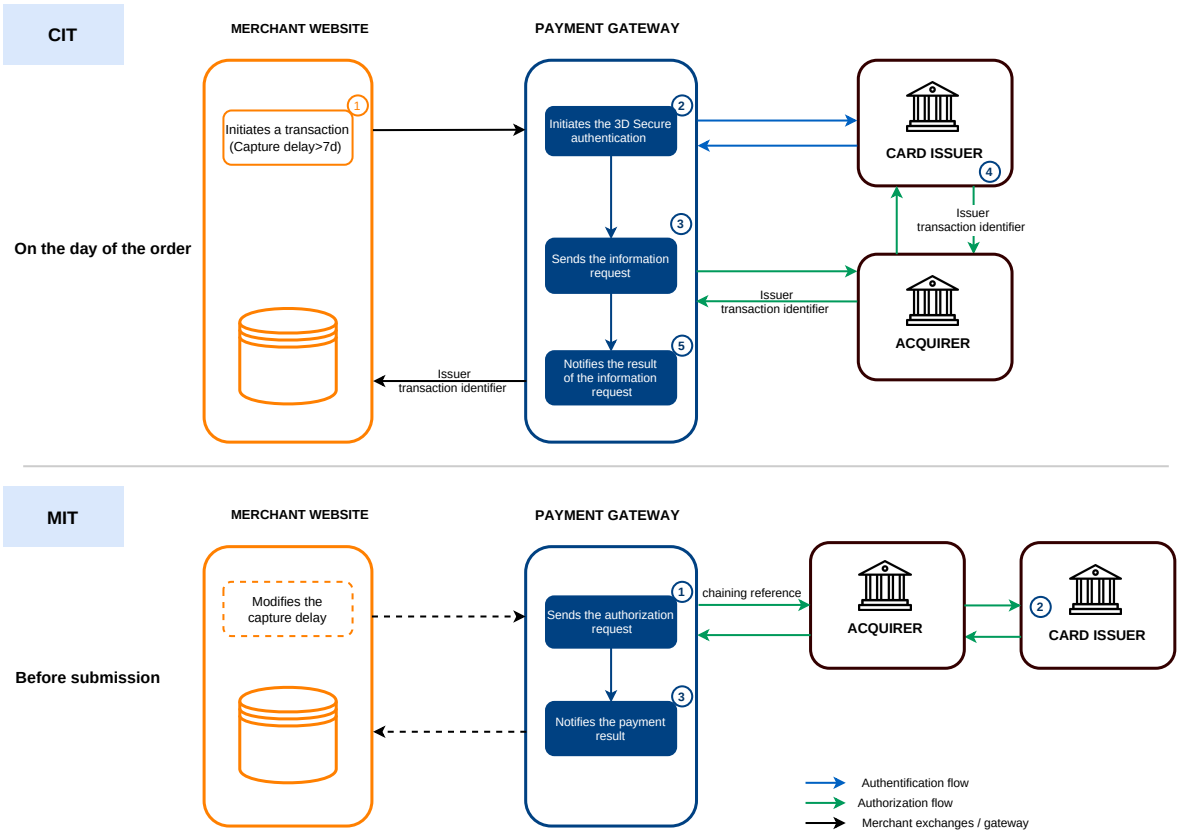
5. The payment gateway notifies the merchant website about the payment result.

Before submission:

1. If the transaction is submitted before the initial capture delay expires, the merchant modifies the capture date to D.
2. If no action is taken on the transaction, the transaction is captured by the bank on the initially requested date.

3.2.2. Capture delay longer than the authorization validity period

Simplified diagram



On the day of the order:

1. The merchant site submits a payment request.
2. The payment gateway initiates the cardholder's authentication process with the issuer.

> The regulation imposes cardholder authentication for all ***CIT*** transactions.

3. Once the authentication (challenge or frictionless) is completed, the gateway proceeds to an information request by providing the cardholder's authentication details.
4. The issuer generates a unique transaction identifier and transmits it in the response to the registration request.
5. The payment gateway notifies the merchant website about the information request result.

Before submission:

1. If no action is taken for the transaction, the authorization request is made on D-1 before the requested capture date.

If it is sent before the initial capture delay expires, the merchant modifies the capture date to D.

The payment gateway performs an authorization request, providing the initial transaction identifier (ITC) as a chaining reference.

2. The issuer recognizes the transaction as an ***MIT*** that is part of a series of payments for which the cardholder has previously authenticated themselves.

   The transaction will not be rejected for lack of authentication (soft decline).

3. If the merchant has enabled the **Instant Payment Notification URL on batch authorization** notification rule, gateway notifies the merchant site of the payment result.

3.3. Payment in installments

The activation of the payment in installments feature is subject to the prior agreement of Société Générale.

A payment is considered to be an “installment payment” if the amount for the purchase is debited to the buyer’s account in several installments.

The first installment works the same way as an immediate payment.

The next installment(s) is similar to (a) deferred payment(s).

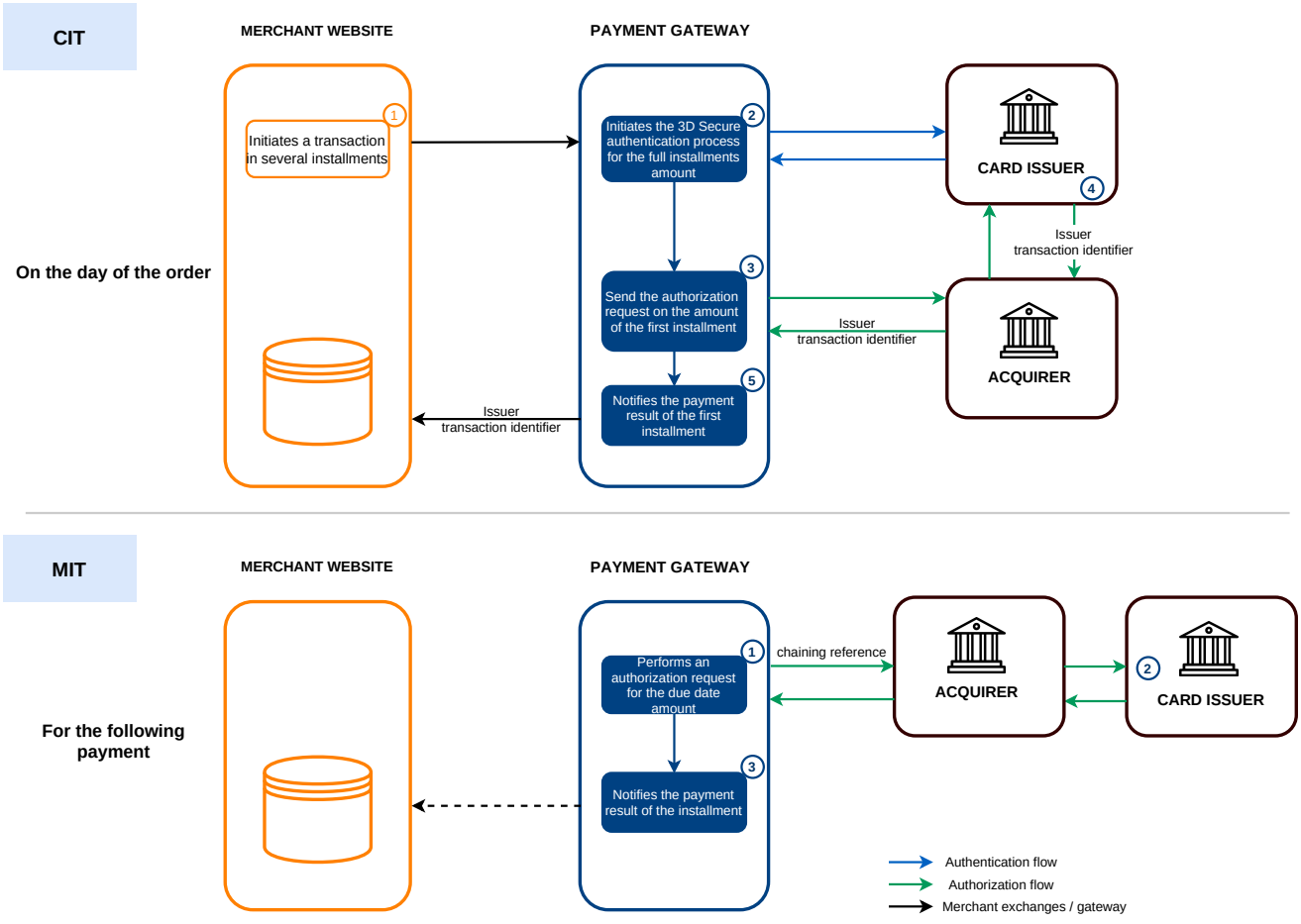
Only the first installment can be guaranteed to the merchant on the condition that the requested capture date for the first installment is set before the authorization expiry date depending on the payment method (see: *Authorization request validity period* on page 22) .

As part of the implementation of PSD2, the cardholder will be required to undergo strong authentication when making the first installment payment.

If the authorization (or information) request is accepted on the day of the order, a transaction is created for each Installment payment due date.

Otherwise, only one rejected transaction is created. The transaction **History** tab then indicates the number of installments initially planned.

Simplified diagram



On the day of the order:

1. The merchant site submits a payment request in several installments.

2. The payment gateway initiates the cardholder's authentication process with the issuer.

   - **Authentication is requested for the total amount of installment.**
   - Regulations require strong authentication for this use case.

3. Once the strong authentication is completed, the gateway proceeds with the authorization request on the amount of the first installment by providing the cardholder's authentication details.
4. The issuer generates a unique transaction identifier and transmits it in the response to the authorization request.
5. The payment gateway notifies the merchant website about the payment result.

For the following payment:

1. The payment gateway performs an authorization request for the installment amount, providing the initial transaction identifier (ITC) as a chaining reference.
2. The issuer recognizes the transaction as a ***MIT*** that is part of a series of payments for which the cardholder has previously authenticated themselves and proceeds with the authorization request.

   The transaction will not be rejected for lack of authentication (soft decline).
3. If the merchant has enabled the **Instant Payment Notification URL on batch authorization** notification rule, gateway notifies the merchant site of the payment result.

In this use case, the way the chaining reference is handled is transparent to the merchant.

Note:

The validity of the payment method is checked throughout the payment schedule.

Otherwise, a warning is displayed to the buyer.

The buyer will then have to specify another payment method or abandon the payment.

It is possible, however, that the card is renewed or canceled before the end of the payment schedule.

In this case, the payments will be refused by the issuing bank (auto 54 return code: Payment method expired.

You will be notified by e-mail about a rejected installment payment via the notification rule "Refusal e-mail for deferred payment".

List of payment methods compatible with payment in installments:

Network code	Payment method	Card types (vads_payment_cards)	Supports payment in installments
ACCORD	Illicado gift Card	ILLICADO	✗
ACCORD	PicWic brand card	PICWIC	✗
ACCORD_SANDBOX	Illicado gift cards - Sandbox mode	ILLICADO_SB	✗
ACCORD_SANDBOX	PicWic brand card - Sandbox mode	PICWIC_SB	✗
AMEXGLOBAL	American Express	AMEX	✓
AURORE	Cpay card	AURORE-MULTI	✗
CB	CB	CB	✓
CB	e-Carte Bleue virtual card	E-CARTEBLEUE	✓

Network code	Payment method	Card types (vads_payment_cards)	Supports payment in installments
CB	Maestro	MAESTRO	✓
CB	Mastercard	MASTERCARD	✓
CB	Visa	VISA	✓
CB	Visa Electron	VISA_ELECTRON	✓
CB	VPay	VPAY	✓
CB	Apetiz Meal Voucher card	APETIZ	✓
CB	Chèque Déjeuner Meal Voucher card	CHQ_DEJ	✓
CB	1st generation Mastercard electronic meal voucher	EDENRED	✓
CB	Sodexo Meal Voucher card	SODEXO	✓
CONECS	Bimpli Meal Voucher card (ex Apetiz)	APETIZ	✗
CONECS	Chèque Déjeuner Meal Voucher card	CHQ_DEJ	✗
CONECS	Conecs Meal Voucher card	CONECS	✗
CONECS	Sodexo Meal Voucher card	SODEXO	✗
CVCONNECT	Chèque-Vacances Connect	CVCO	✗
FRANFINANCE	Franfinance payment in 3X	FRANFINANCE_3X	✗
FRANFINANCE	Franfinance payment in 4X	FRANFINANCE_4X	✗
FRANFINANCE_SB	Franfinance payment in 3X - Sandbox mode	FRANFINANCE_3X	✗
FRANFINANCE_SB	Franfinance payment in 4X - Sandbox mode	FRANFINANCE_4X	✗
FULLCB	Payment in 3 installments with no fees with BNPP PF	FULLCB3X	✗
FULLCB	Payment in 4 installments with no fees with BNPP PF	FULLCB4X	✗
MASTERPASS	MasterPass	MASTERPASS	✗
ONEY_API	Oney 3x 4x payment	ONEY_3X_4X	✗
ONEY_API	Payment 10x 12x Oney	ONEY_10X_12X	✗
ONEY_API	Payment Oney Pay Later	ONEY_PAYLATER	✗
ONEY_API	Oney partner brand cards	ONEY_ENSEIGNE	✗
ONEY_API_SANDBOX	Oney 3x 4x payment (Sandbox mode)	ONEY_3X_4X	✗
ONEY_API_SANDBOX	Oney 10x 12x payment (Sandbox mode)	ONEY_10X_12X	✗
ONEY_API_SANDBOX	Payment Oney Pay Later (Sandbox mode)	ONEY_PAYLATER	✗
ONEY_API_SANDBOX	Oney partner brand cards in Sandbox mode	ONEY_ENSEIGNE	✗
PAYPAL	PayPal	PAYPAL	✗
PAYPAL_SB	PayPal - Mode sandbox	PAYPAL_SB	✗
PLANET_DCC	MASTERCARD	MASTERCARD	✓
PLANET_DCC	VISA	VISA	✓

Network code	Payment method	Card types (vads_payment_cards)	Supports payment in installments
SEPA	SEPA DIRECT DEBIT	SDD	✗

3.4. Cascading payment

Cascading payment is a service allowing the buyer to pay an entire order using several payment methods.

Only “prepaid card”-type payment methods can be used for the cascading payment.

Three possible use cases:

- If the prepaid card balance is less than the order amount, the payment is automatically split.
- Prepaid cardholders can choose to split their payment by paying part of the order with their card and the rest with another compatible payment method.
- The payment platform enables the cardholder to make a cascading payment using only several prepaid cards, without using a bank card. Example: pay for part of the order with an Illicado gift card and the balance with an Auchan gift card.

Use in a payment form

Use the vads_payment_cards form field to list the payment methods to be proposed.

- Example of 3 payment methods:

```
vads_payment_cards="VISA;MASTERCARD;AUCHAN_SB"
```

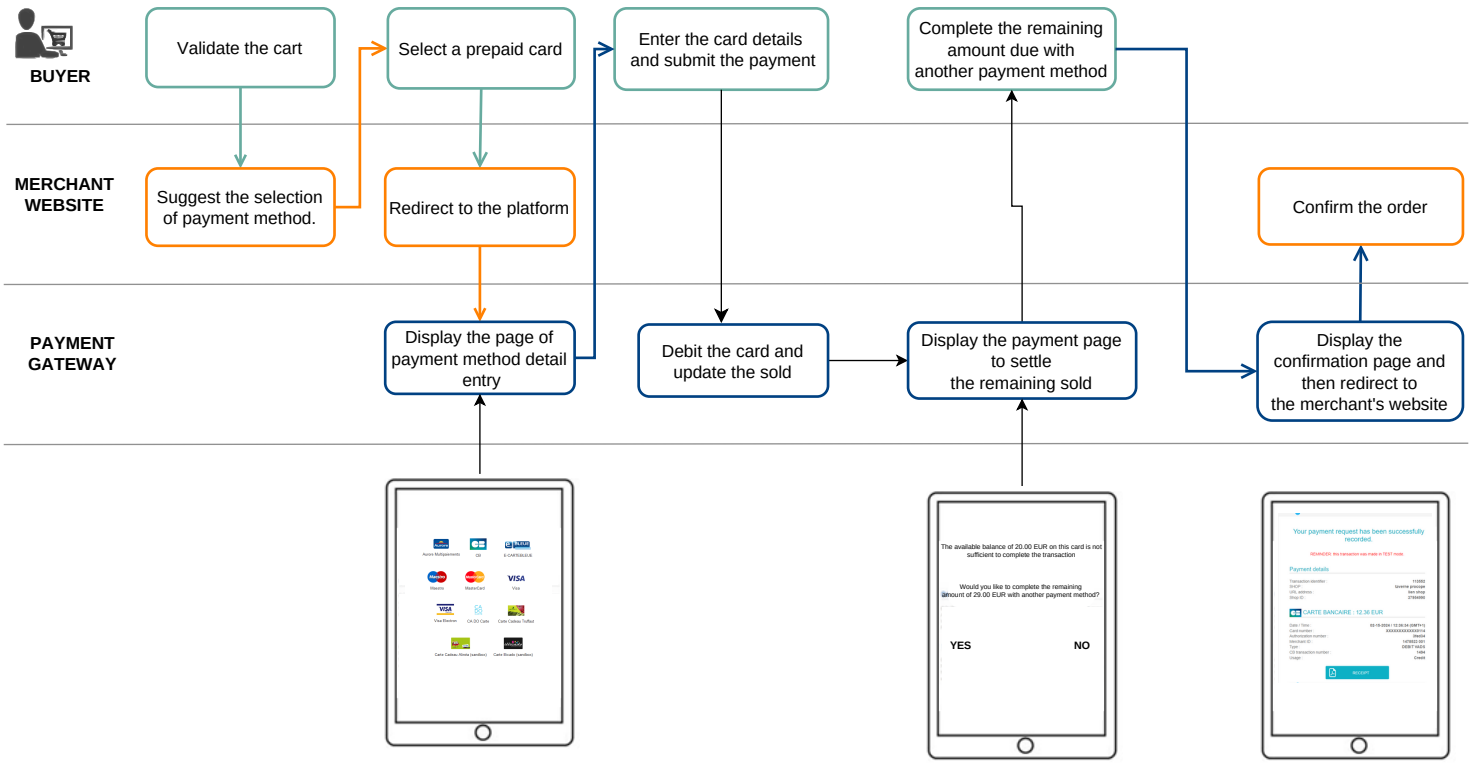
- Example of 4 payment methods:

```
vads_payment_cards="VISA;MASTERCARD;AUCHAN_SB;CVCO"
```

If the vads_payment_cards field is not sent, or if it is sent empty, then all eligible payment methods associated with the shop will be offered.

Payment process of a cascading payment

Example of payment: the buyer uses a gift card whose balance is less than the total amount of the order.



List of payment methods compatible with cascading payment:

- Sign cards

Payment method	Card types (vads_payment_cards)
Accord brand card	ACCORD_STORE
Alinéa brand card	ALINEA
Auchan brand card	AUCHAN
Boulanger brand card	BOULANGER
Leroy-Merlin brand card	LEROY-MERLIN
Norauto brand card	NORAUTO
PicWic brand card	PICWIC
Villaverde brand card	VILLAVERDE
Accord brand card - Sandbox mode	ACCORD_STORE_SB
Auchan brand card - Sandbox mode	AUCHAN_SB
Boulanger brand card - Sandbox mode	BOULANGER_SB
Leroy-Merlin brand card - Sandbox mode	LEROY-MERLIN_SB
Norauto brand card - Sandbox mode	NORAUTO_SB
PicWic brand card - Sandbox mode	PICWIC_SB
Villaverde brand card - Sandbox mode	VILLAVERDE_SB

- Gifts cards

Payment method	Card types (vads_payment_cards)
Alinéa gift card	ALINEA_CDX
Allobébé gift card	ALLOBEBE_CDX
BizzBee gift card	BIZZBEE_CDX
Brice gift card	BRICE_CDX
Illicado gift Card	ILLICADO
Jouéclub gift card	JOUECLUB_CDX
Allobébé gift card - Sandbox mode	ALLOBEBE_CDX_SB
BizzBee gift card - Sandbox mode	BIZZBEE_CDX_SB
Brice gift card - Sandbox mode	BRICE_CDX_SB

Payment method	Card types (vads_payment_cards)
Illicado gift card - Sandbox mode	ILLICADO_SB
JouéClub gift card - Sandbox mode	JOUECLUB_CDX_SB

- Meal Voucher cards

Payment method	Card types (vads_payment_cards)
Bimpli Meal Voucher card (ex Apetiz)	APETIZ
Chèque Déjeuner Meal Voucher card	CHQ_DEJ
Conecs Meal Voucher card	CONECS
Sodexo Meal Voucher card	SODEXO
EDENRED Meal Voucher card	EDENRED

- Chèque-Vacances Connect

Payment method	Card types (vads_payment_cards)
Chèque-Vacances Connect	CVCO

3.5. Offering payment in a foreign currency

Payment in foreign currency with conversion allows Merchants to present price catalogs in different currencies, but without having to manage their finances in currencies different from the ones specified in their contract.

When the gateway receives the amount in a currency not managed by your MIDs, it makes a conversion to the company's currency based on the daily exchange rate provided by Visa.

The buyer is informed of the indicative rate at the time of payment, but does not really know the final amount of the transaction.

The capture at the bank does not necessarily occur on the day of the authorization and the rate may therefore vary between the date of authorization and the date of capture.

For this reason, the counter-value displayed at the time of payment is provided as an indication.

- The authorization request is sent in the currency of the contract to the card issuer.
- The capture is performed exclusively in the currency of the contract.
- The buyer is debited in the contract currency with exchange fees applied by their bank, without managing the exchange rate.

At the end of the payment, the merchant receives a notification containing the following fields:

- **vads_amount**: the currency amount,
- **vads_currency**: the currency,
- **Vads_effective_amount**: the actual amount in the currency of their contract, calculated using the exchange rate in force at the time of the authorization,
- **vads_effective_currency**: the currency that will be used for the capture,
- **vads_change_rate**: the exchange rate applied for converting the amount in the currency of the contract to the buyer's currency.

3.6. The “Anticipated authorizations” service

This service allows to trigger the authorization on D-Δ (see *Authorization validity period* for each payment method) before the desired capture date at the bank.

In case of refusal by the issuing bank, exclusively for a non-fraud related reason, a process automatically reissues authorization requests until up to 2 days prior to the desired capture date at the bank.

The merchant may cancel the transaction or change its amount (only smaller amounts can be entered) and/or the capture date at any moment.

This process applies to:

- recurring payments,
- deferred payments,
- installments, other than the first one, in case of payment in installments.

In case of refusal for fraud-related reasons, the transaction is considered as permanently rejected.

Here is a list of fraud-related reasons that do not allow authorization reruns.

Network	Authorization return code	Label
CB	03	Invalid acceptor
	04	Keep the card
	05	Do not honor
	07	Keep the card, special conditions
	12	Incorrect Transaction Code
	13	Invalid amount
	14	Invalid cardholder number
	15	Unknown issuer
	31	Unknown acquirer company ID
	33	Expired card
	34	Suspected fraud
	41	Lost card
	43	Stolen card
	54	Expired card
	56	Card absent from the file
	57	Transaction not allowed for this cardholder
	59	Transaction not allowed for this cardholder
	63	Security rules unfulfilled
	76	The cardholder is already blocked, the previous record has been saved
	80	Contactless payment is not accepted by the issuer
	81	Unsecured payment is not accepted by the issuer
	82	Revocation of recurring payment for the card of a specific Merchant or for the MCC and the card
	83	Revocation of all recurring payments for the card

Contact your customer advisorSociété Générale if you would like to enable anticipated authorizations.

3.7. Authorization request validity period

Network code	Payment method	Card types (vads_payment_cards)	Authorization validity period (in days)
ACCORD	Illicado gift Card	ILLICADO	0
ACCORD	PicWic brand card	PICWIC	0
ACCORD_SANDBOX	Illicado gift cards - Sandbox mode	ILLICADO_SB	0
ACCORD_SANDBOX	PicWic brand card - Sandbox mode	PICWIC_SB	0
AMEXGLOBAL	American Express	AMEX	7
AURORE	Cpay card	AURORE-MULTI	29
CB	CB	CB	7
CB	e-Carte Bleue virtual card	E-CARTEBLEUE	7
CB	Maestro	MAESTRO	30
CB	Mastercard	MASTERCARD	7
CB	Visa	VISA	7
CB	Visa Electron	VISA_ELECTRON	7
CB	VPay	VPAY	7
CB	Bimpli Meal Voucher card (ex Apetiz)	APETIZ	7
CB	Chèque Déjeuner Meal Voucher card	CHQ_DEJ	7
CB	1st generation Mastercard electronic meal voucher	EDENRED	7
CB	Sodexo Meal Voucher card	SODEXO	7
CONECS	Bimpli Meal Voucher card (ex Apetiz)	APETIZ	30
CONECS	Chèque Déjeuner Meal Voucher card	CHQ_DEJ	30
CONECS	Conecs Meal Voucher card	CONECS	30
CONECS	Sodexo Meal Voucher card	SODEXO	30
CVCONNECT	Chèque-Vacances Connect	CVCO	6
FRANFINANCE	Franfinance payment in 3X	FRANFINANCE_3X	0
FRANFINANCE	Franfinance payment in 4X	FRANFINANCE_4X	0
FRANFINANCE_SB	Franfinance payment in 3X - Sandbox mode	FRANFINANCE_3X	0
FRANFINANCE_SB	Franfinance payment in 4X - Sandbox mode	FRANFINANCE_4X	0
FULLCB	Payment in 3 installments with no fees with BNPP PF	FULLCB3X	7
FULLCB	Payment in 4 installments with no fees with BNPP PF	FULLCB4X	7
MASTERPASS	MasterPass	MASTERPASS	0
ONEY_API	Oney 3x 4x payment	ONEY_3X_4X	0
ONEY_API	Payment 10x 12x Oney	ONEY_10X_12X	0
ONEY_API	Payment Oney Pay Later	ONEY_PAYLATER	0
ONEY_API	Oney partner brand cards	ONEY_ENSEIGNE	0
ONEY_API_SANDBOX	Oney 3x 4x payment (Sandbox mode)	ONEY_3X_4X	0
ONEY_API_SANDBOX	Oney 10x 12x payment (Sandbox mode)	ONEY_10X_12X	0
ONEY_API_SANDBOX	Payment Oney Pay Later (Sandbox mode)	ONEY_PAYLATER	0
ONEY_API_SANDBOX	Oney partner brand cards in Sandbox mode	ONEY_ENSEIGNE	0

Network code	Payment method	Card types (vads_payment_cards)	Authorization validity period (in days)
PAYPAL	PayPal	PAYPAL	3
PAYPAL_SB	PayPal - Mode sandbox	PAYPAL_SB	3
PLANET_DCC	MASTERCARD	MASTERCARD	0
PLANET_DCC	VISA	VISA	0
SEPA	SEPA DIRECT DEBIT	SDD	15

4. THE 3D SECURE AUTHENTICATION

You can find all useful information about 3DS authentication in the *3D Secure guide*.

4.1. “Frictionless” flow

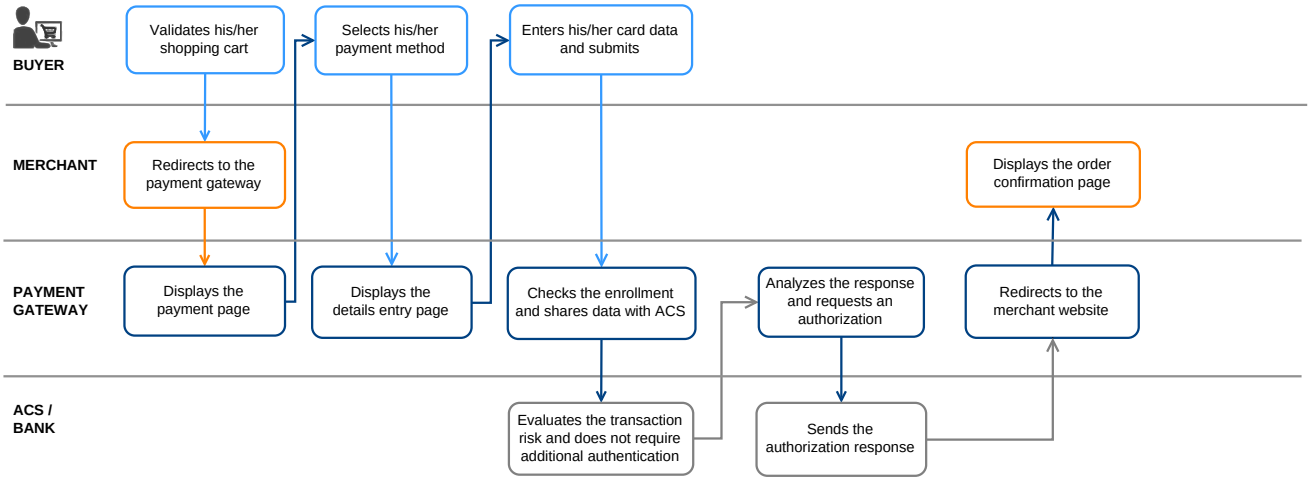
In frictionless flow (without interaction with the buyer), based on the received information, the issuer can determine:

- That no additional authentication is required.

  The payment gateway proceeds with the payment and issues the authorization request.

- That the analyzed information does not provide the authorization to proceed with the payment.

  In this case, the payment gateway notifies the merchant website and the buyer about the payment rejection and redirects the buyer to the merchant website.



4.2. “Challenge” flow

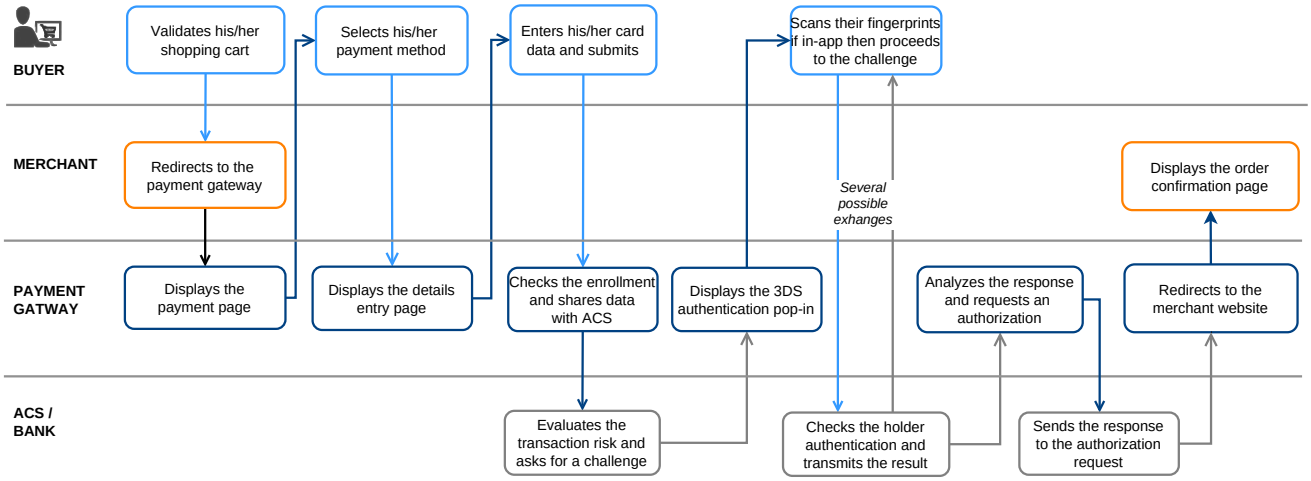
In a challenge flow, based on the received information, the issuer determines that it is necessary for the buyer to provide the following elements:

- Either a biometric element, such as a device fingerprint,
- or a strong authentication via two-factor authentication.

For in-app solutions, the device fingerprint will be systematically requested before proceeding to the challenge.

Once the challenge has been successfully completed, the payment gateway proceeds with the payment and issues the authorization request.

In case of a technical or authentication error, the payment stops. The payment gateway notifies the merchant website and the buyer about the payment rejection and redirects the buyer to the merchant website.



4.3. Increasing the chances of a frictionless payment

- The use of these fields is optional. In any case, it is the issuing bank that decides if strong authentication must be performed.

Name/Description	Format/Values
vads_cust_address_number Street number - Billing address	Format: ans..64
vads_cust_address2 2nd line of the address - Billing address	Format: ans..255
vads_cust_address 1st line of the address - Billing address	Format: ans..255
vads_cust_cell_phone Buyer's cell phone number	Format: an..32
vads_cust_city City - Billing address	Format: an..128
vads_cust_email Cardholder's e-mail address	Format: ans..150
vads_cust_national_id National identifier. Allows to identify each citizen of a country in a unique way	Format: ans..255
vads_cust_phone Shipping buyer's phone number	Format: an..32
vads_cust_state State/Region - Billing address	Format: ans..127
vads_cust_zip Zip code - Billing address	Format: an..64
vads_ship_to_city City - Shipping address	Format: an..128
vads_ship_to_email Shipping e-mail address in case of an e-ticket order.	Format: an..128
vads_ship_to_type Transport type	Format: enum 3DS2 value: • **"CARD_HOLDER_ADDRESS"** • **"VERIFIED_ADDRESS"** • **"NOT_VERIFIED_ADDRESS"** • **"SHIP_TO_STORE"** • **"DIGITAL_GOOD"** • **"ETRAVEL_OR_ETICKET"** • **"OTHER"** • **"PICKUP_POINT"** • **"AUTOMATED_PICKUP_POINT"**
vads_ship_to_state State/Region - Shipping address	Format: ans..127
vads_ship_to_street2 2nd line of the address - Shipping address	Format: ans..255
vads_ship_to_street 1st line of the address - Shipping address	Format: ans..255
vads_ship_to_speed Shipping speed	Format: enum 3DS2 value: • **"ELECTRONIC_DELIVERY"**

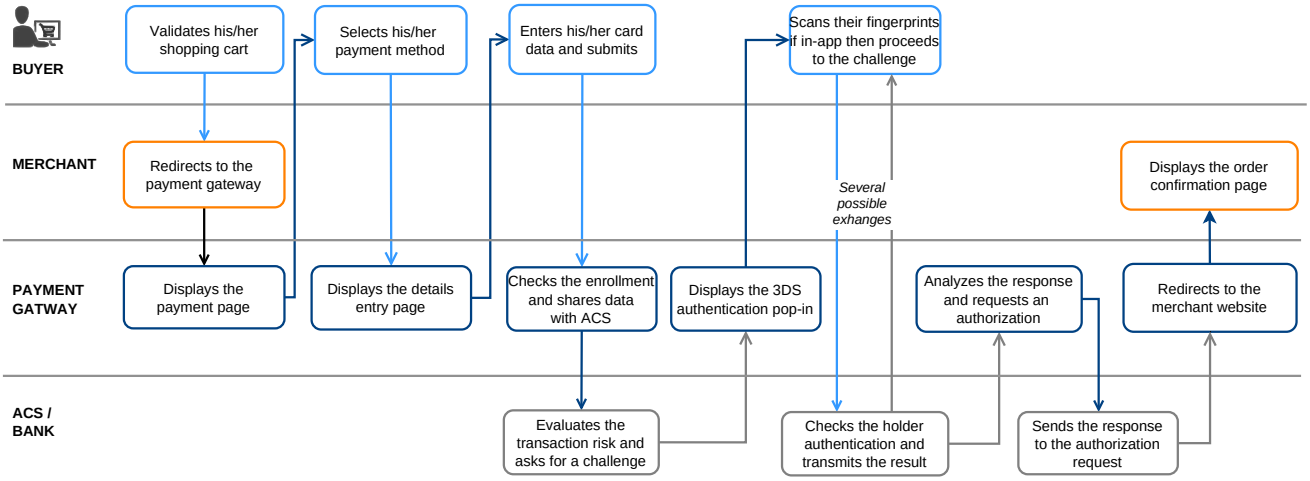
Name/Description	Format/Values
	• "**SAME_DAY_SHIPPING**" • "**OVERNIGHT_SHIPPING**" • "**TWO_DAYS_OR_MORE_SHIPPING**"
vads_ship_to_zip Zip code - Shipping address	Format: ans..64

5. UNDERSTANDING THE PAYMENT FLOW

The online payment process appears differently when viewed from the point of view of the buyer or of the merchant.

5.1. Defining the steps of the payment process - As seen by the buyer

Payment process from the buyer's point of view :



1. The buyer validates the shopping cart.
2. The merchant website redirects the buyer to the payment gateway.

   This redirection is done via an HTML POST form in 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, presents the payment process to the buyer.

   There are two journey, depending on how you fill in the payment form:

   - **Journey 1**: a single payment method is specified in the payment form (by credit card only, for example).

     In this step, the platform directly displays the *Payment method data entry page (step 5)*.
   - **Journey 2**: several payment methods are offered in the payment form (credit card and other payment methods available in the store).

     The platform then displays the payment method selection page. Example:

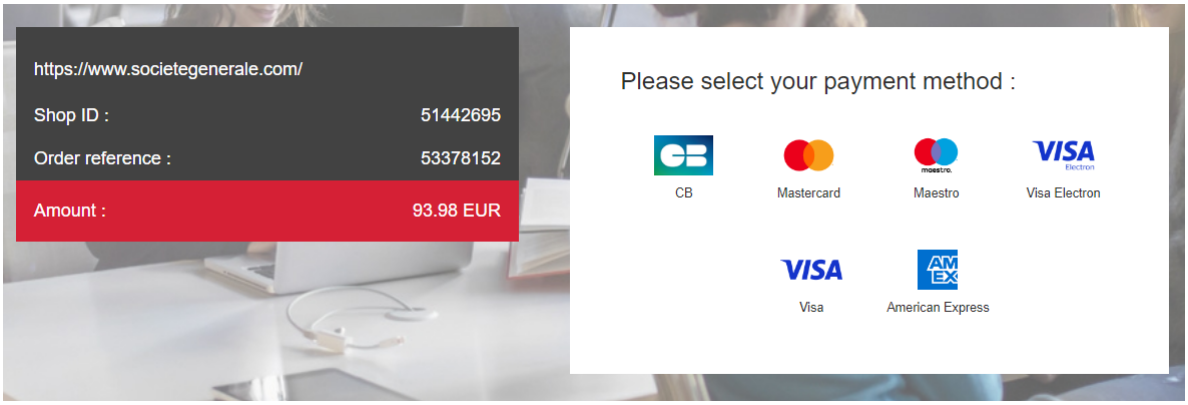


Figure 1: Selecting a payment method

CB, Visa and Mastercard are grouped together under a single logo.

This feature allows the buyer to go directly to the card data entry page if the merchant has only one CB contract.

If, in addition to the CB contract, the merchant has several other payment methods, then CB, e-Carte bleue, Visa, Visa Electron, Mastercard and Maestro are grouped together under a single logo when the payment page is displayed.

4. The buyer selects his payment method if the platform displays rjourney 2.
5. The buyer enters the number and the expiry date of their card.

   If the card has a security (CVV) code, it must be specified.

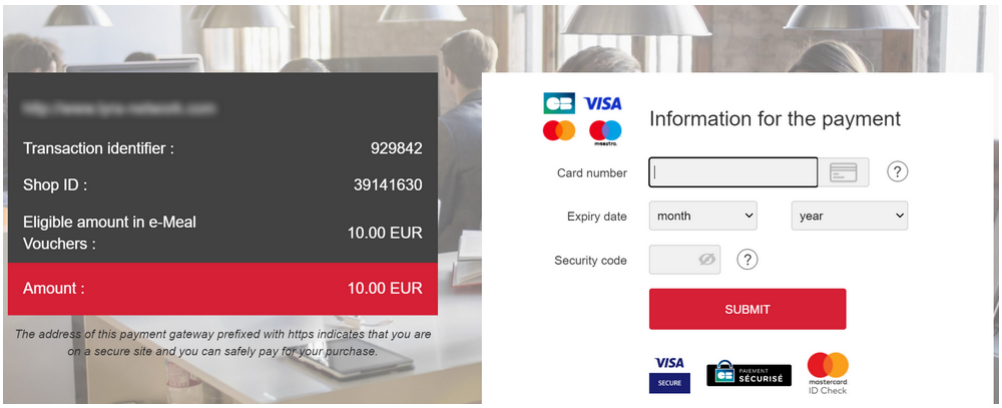


Figure 2: Entering payment method details

6. When paying with a CB co-badged Visa or Mastercard card, the payment platform automatically detects the brand(s) associated with the card number entered.

   If several brands are available, a drop-down list appears to the right of the entry field. The logo chosen by the merchant appears first on the list.

   If only one brand is available, the logo appears automatically.

   *If you need help, you can click the **?** icon to the right of the entry field.*
7. The buyer click on **Validate**..
8. If the merchant and the buyer's card are enrolled in the 3D Secure program, the payment will be authenticated with 3D Secure.
9. The platform requests authorization from the buyer's bank, the issuer, in addition to the payment platform's internal fraud checks.

10. If successful, the platform presents the buyer with a summary page summarizing the transaction information.

    The logo of the brand selected by the buyer appears on the receipt.

    A button allowing to return to the shop is presented.

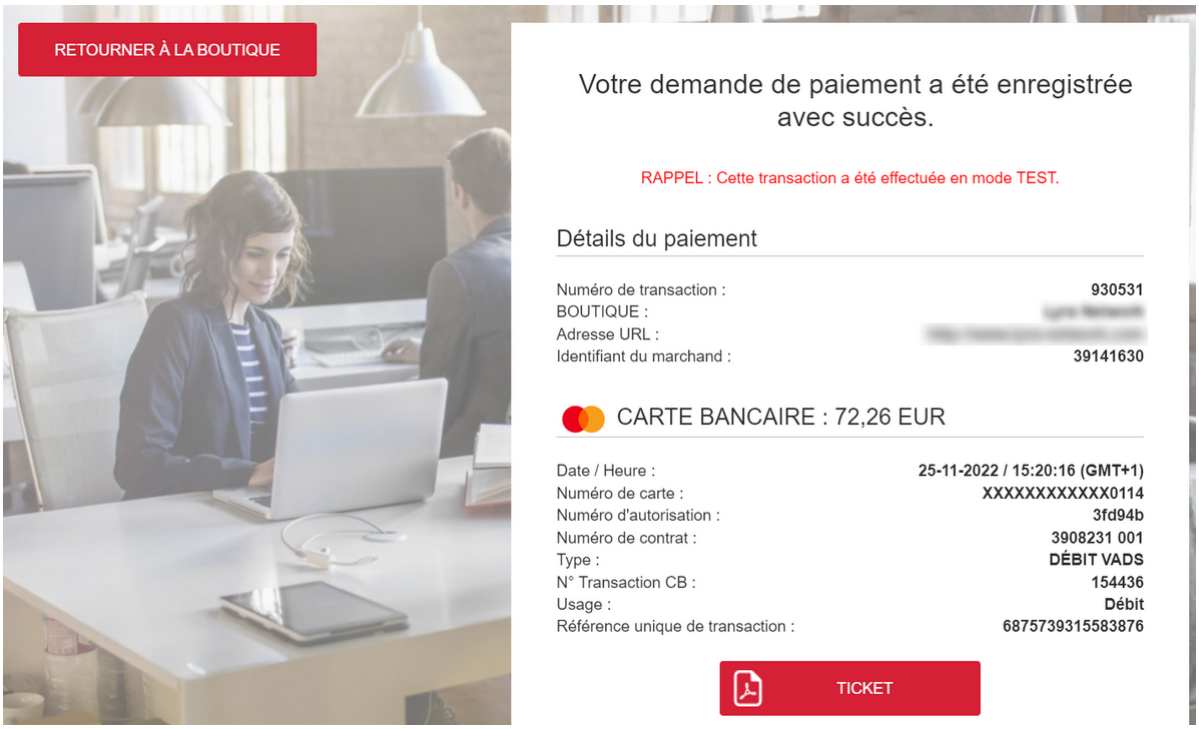


Figure 3: Transaction summary

In the event of failure, the platform informs the buyer and displays a button to cancel and return to the shop.

If you have configured a number of additional attempts that is higher than 0 in your Merchant Back Office, the buyer has the opportunity to make another attempt to finalize their payment. If he or she accepts, the payment process resumes at the stage of payment method selection.

Once the additional attempts have been used, the payment is permanently rejected.

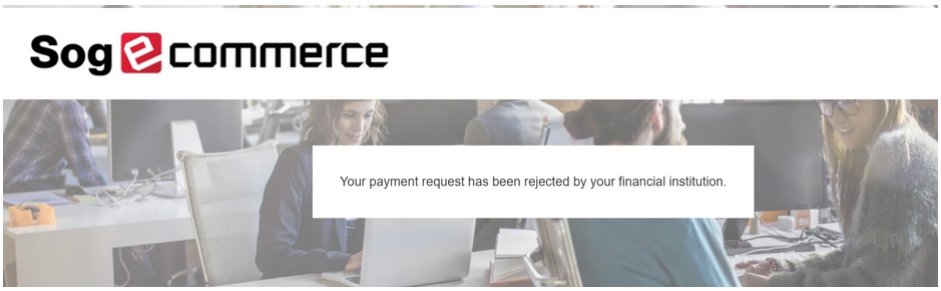


Figure 4: Summary page in case of a failed transaction

5.2. Defining the steps of the payment process - As seen by the merchant

Here is what the online payment process looks like from the merchant's point of view:

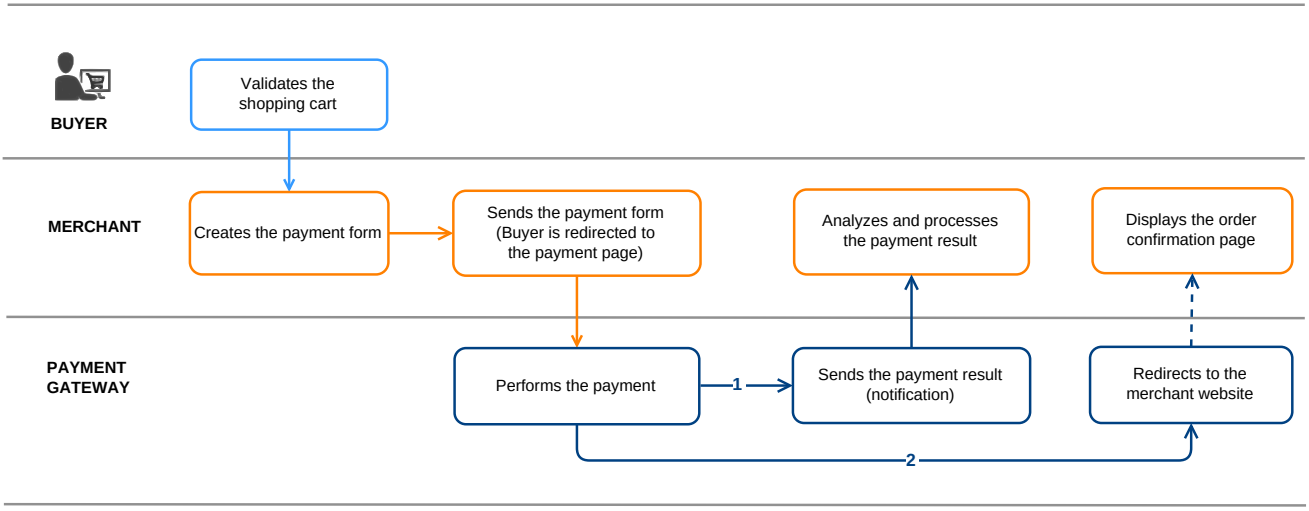


Figure 5: Payment process - as seen by the merchant

1. The buyer validates the shopping cart.
2. The merchant website creates a form using the data from the buyer's cart.
3. 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*.
4. After the buyer enters the payment method details, the payment gateway proceeds to the payment.
5. Depending on the shop configuration (see chapter *Setting up notifications*), the payment gateway calls automatically the merchant website to transmit the result.
6. The merchant website analyzes and processes the payment result.
7. It updates the database (order/stock status, etc.).
8. The buyer sees the payment result on the payment gateway. If the buyer decides to return to the merchant website, he/she sees a "thank you" message and the order status appears.

6. OFFERING ADDITIONAL PAYMENT ATTEMPTS

When a payment is refused, you have the possibility to offer the buyer to make another attempt with another payment method or to re-enter their data in case of a data entry error.

The number of additional attempts can be configured via the Merchant Back Office:

1. Go to **Settings** > **Shop**, then click on the name of the shop for which the configuration must be changed.
2. Select **Configuration**.
3. Enter the authorized number of additional attempts in case of a rejected payment.

   If you configure 2 additional attempts, the buyer will be able to make 3 payment attempts in total.
4. If you wish, you can configure an IPN that will be sent upon each rejected attempt by checking **Instant Payment Notification URL on a declined attempt**.
5. Click **Save**.

Additional attempts are not offered:

- If the payment is an installment payment,

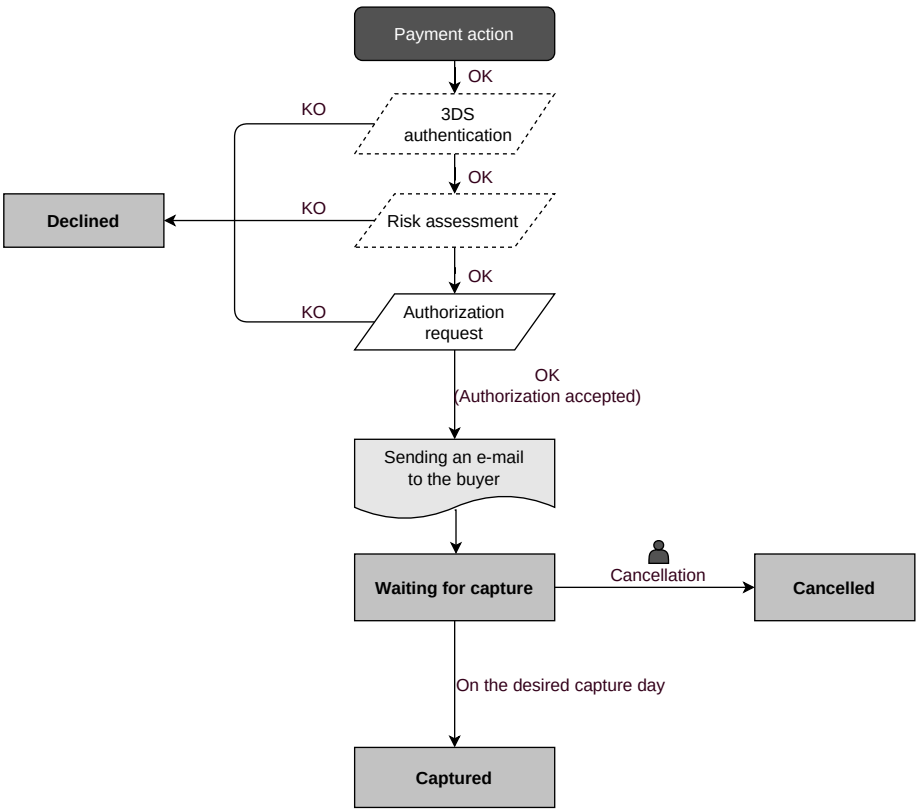
7. TRANSACTION LIFECYCLE

In all the following diagrams, the following caption is used:

Action required from the merchant - manual (Merchant Back Office) or automatic (Web Services)

7.1. Immediate payment

7.1.1. Automatic validation



Once the payment request has been made, several verification processes start automatically:

- The 3D Secure authentication.
- Different verification processes performed by the payment gateway (these potentially include local checks, risk rules configured by the merchant) or by an external risk analyzer.
- An authorization request is also made by the buyer's bank on the day of payment, independently of the requested capture date at the bank.

If one of the verification processes fails, the payment request will not be accepted. The buyer is informed of the rejection on the screen. In the Merchant Back Office, the transaction appears with the **Refused** status.

Otherwise, the transaction takes the **Waiting for capture** status.

The buyer is informed about the acceptance of the payment request and receives a confirmation e-mail.

The transaction will be automatically submitted for capture on the day requested by the merchant and will take the **Captured** status. The **Captured** status is final.

Once the capture is made, the arrival of the transaction to the merchant account depends on the interbank processing time.

Before the capture date, the buyer can modify it together with the amount (only smaller amounts can be entered in case of partial delivery by the merchant).

If necessary, the buyer can also cancel the transaction: the transaction will then appear with the **Cancelled** status.

7.1.2. Manual validation

Following a payment request, the verification process starts automatically:

- The 3D Secure authentication.
- Different verification processes performed by the payment gateway (these potentially include local checks, risk rules configured by the merchant) or by an external risk analyzer.
- An authorization request is also made by the buyer's bank.

If one of the verification processes fails, the payment request will not be accepted. The buyer is informed of the rejection on the screen. In the Merchant Back Office, the transaction appears with the **Refused** status.

Otherwise, the payment is accepted and the transaction appears in the Merchant Back Office with the **To be validated** status.

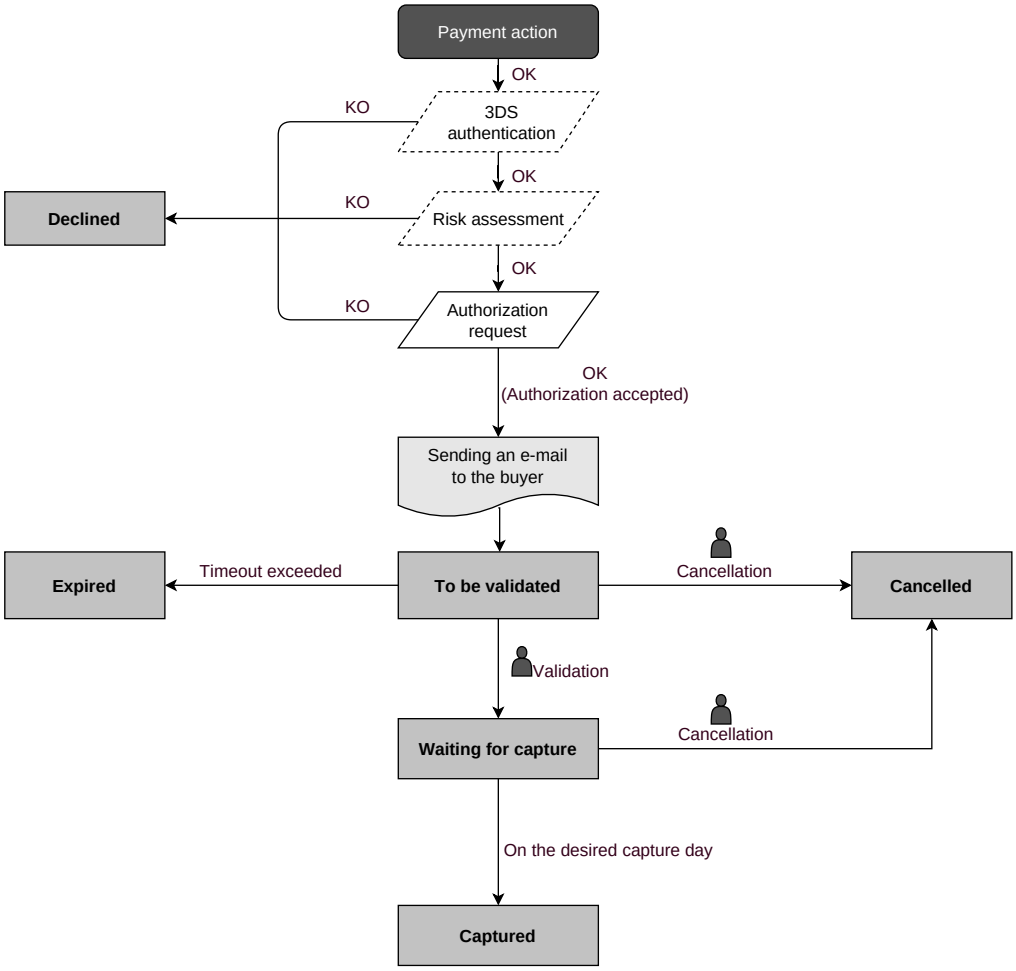
In this case, the merchant must validate the transaction before the expiry date of the authorization request. If the validation is made after this date, the transaction appears as **Expired** and cannot be captured in the bank.

As soon as the transaction is validated, its status changes to **Waiting for capture**.

The transaction will be automatically submitted for capture on the day requested by the merchant and will take the **Captured** status. The **Captured** status is final.

Once the capture is made, the arrival of the transaction to the merchant account depends on the interbank processing time.

The merchant can also cancel the transaction, if necessary. In this case, the transaction takes the **Cancelled** status.



7.2. Deferred payment

7.2.1. Automatic validation

Capture delay shorter than the authorization validity period

(See the diagram “The life cycle of an immediate payment transaction”).

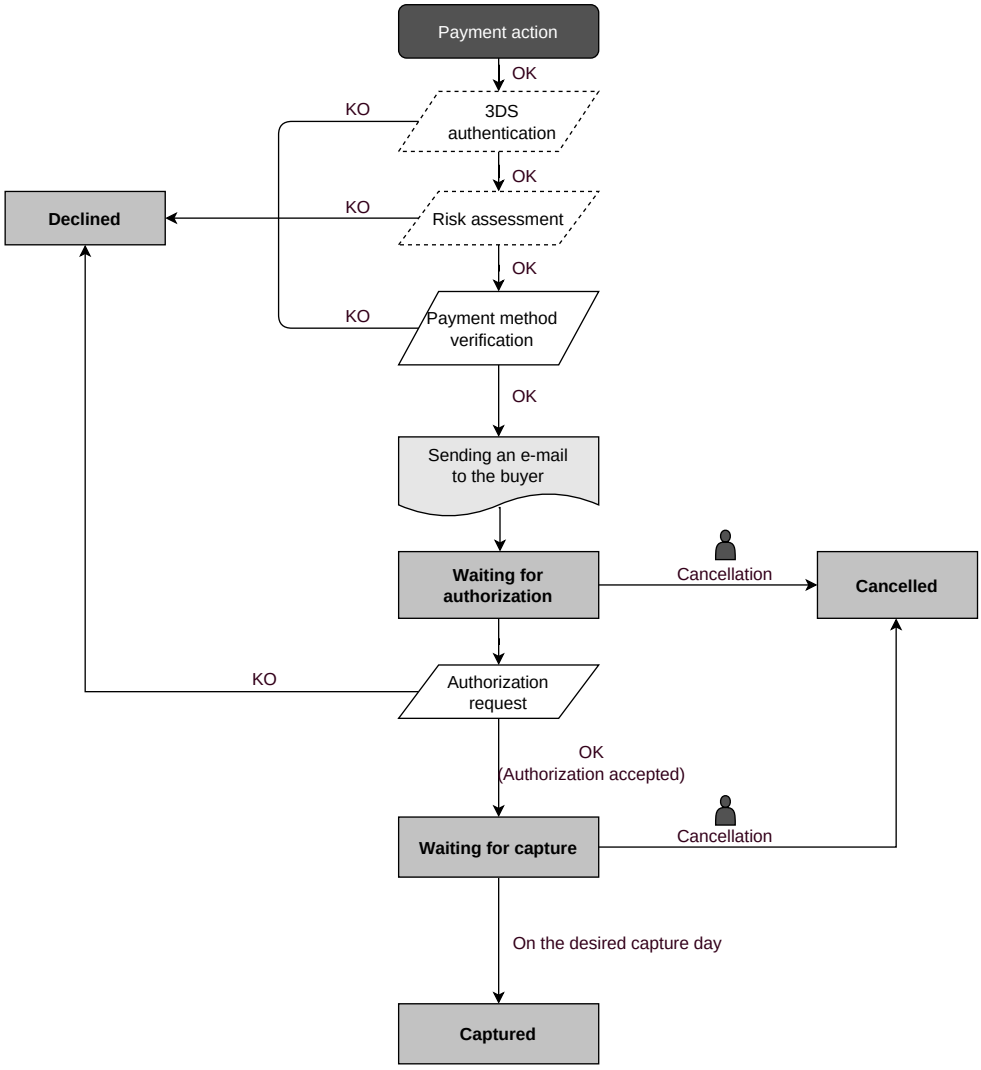
Capture delay longer than the authorization validity period

All the transactions for deferred payments made in automatic validation mode with a successfully completed verification request can be viewed in the Merchant Back Office with the **Waiting for authorization** status.

The authorization request is automatically sent:

- By default: the day before the desired capture date,
- With anticipated authorization: depending on the selected payment method, on D-Δ before the desired capture date (see chapter *The “Anticipated authorizations” service* on page 21).

A deferred payment goes through the steps in the diagram below:



7.2.2. Manual validation

Capture delay shorter than the authorization validity period

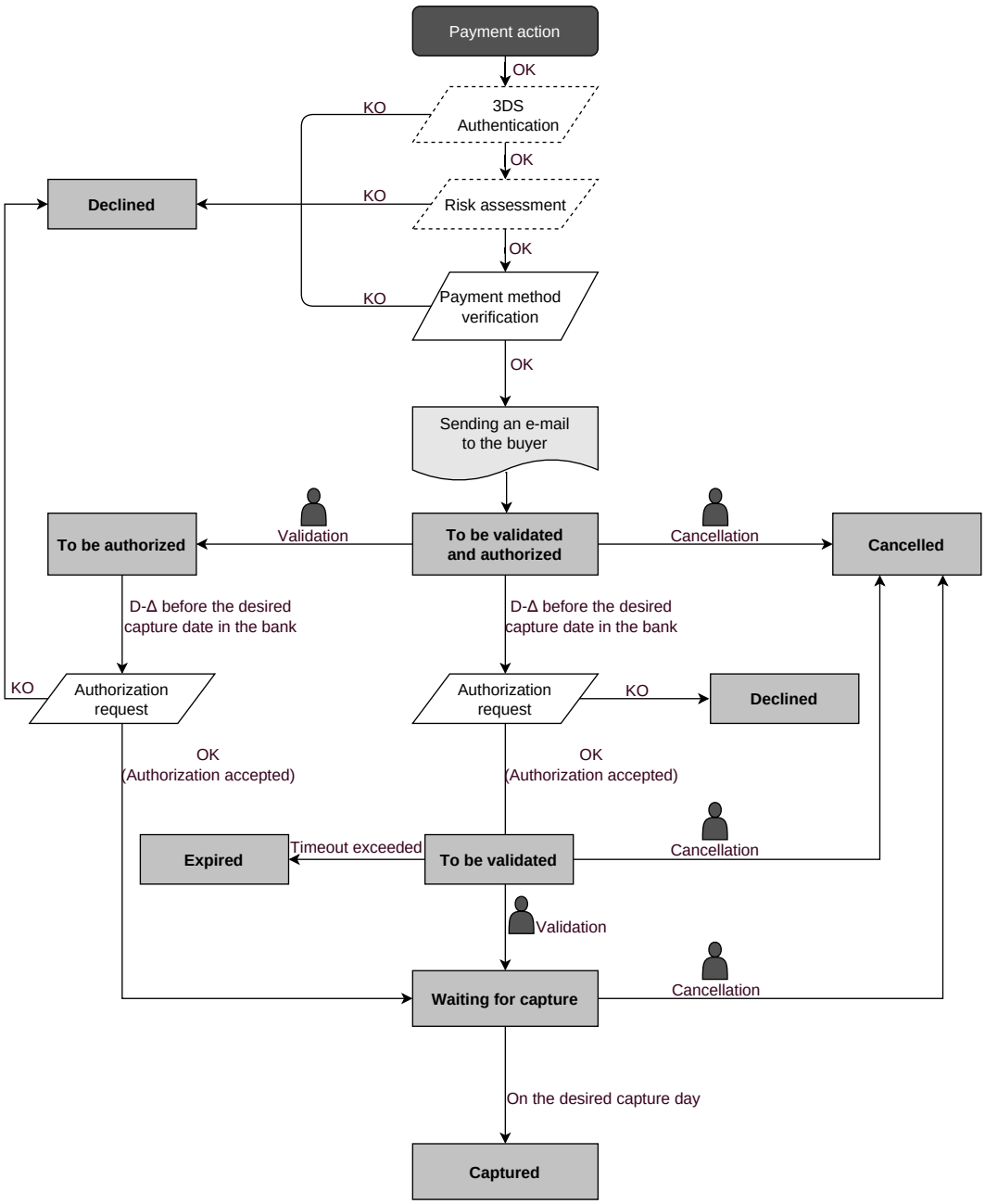
(See the diagram “The life cycle of an immediate payment transaction”).

Capture delay longer than the authorization validity period

All the transactions for deferred payments made in automatic validation mode with a successfully completed authorization request for EUR 1 (or information request about the CB network if the acquirer supports it) can be viewed in the Merchant Back Office with the **To be validated and authorized** status.

The authorization request is automatically sent on the requested capture day, on the condition that the merchant has already validated the transaction.

In the meantime, the merchant may cancel the transaction or change its amount (only smaller amounts can be entered) and/or the capture date. These transactions go through the steps in the diagram below:



7.3. Payment in installments

7.3.1. Automatic validation

The activation of the payment in installments feature is subject to the prior agreement of Société Générale.

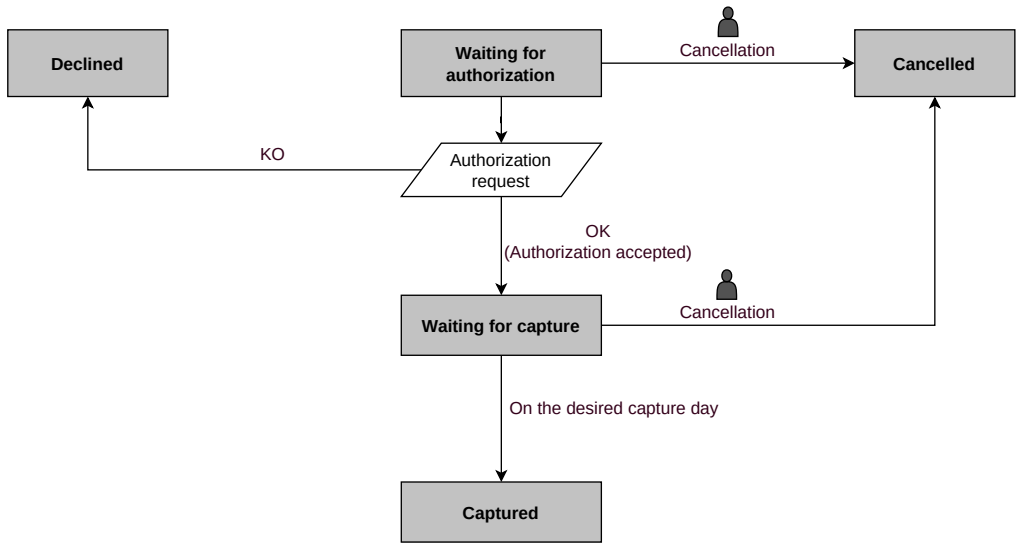
Depending on the capture date, the first installment will have exactly the same features as an immediate payment or a deferred payment.

By default, the following installments will have the **Waiting for authorization** status. The buyer's bank will be able to reject the authorization request. The payment gateway will then inform the merchant by e-mail that the transaction has been declined.

The authorization requests for the upcoming installments are automatically sent as a transaction for a deferred payment, with two possible dates:

- By default: the day before the desired capture date,
- With anticipated authorization: depending on the selected payment method, on D-Δ before the desired capture date (see chapter *The "Anticipated authorizations" service* on page 21).

The following installments go through the steps specified in the diagram below (case of an authorization request that is not resent):



In any case, canceling an installment never implies that the upcoming installments will be canceled.

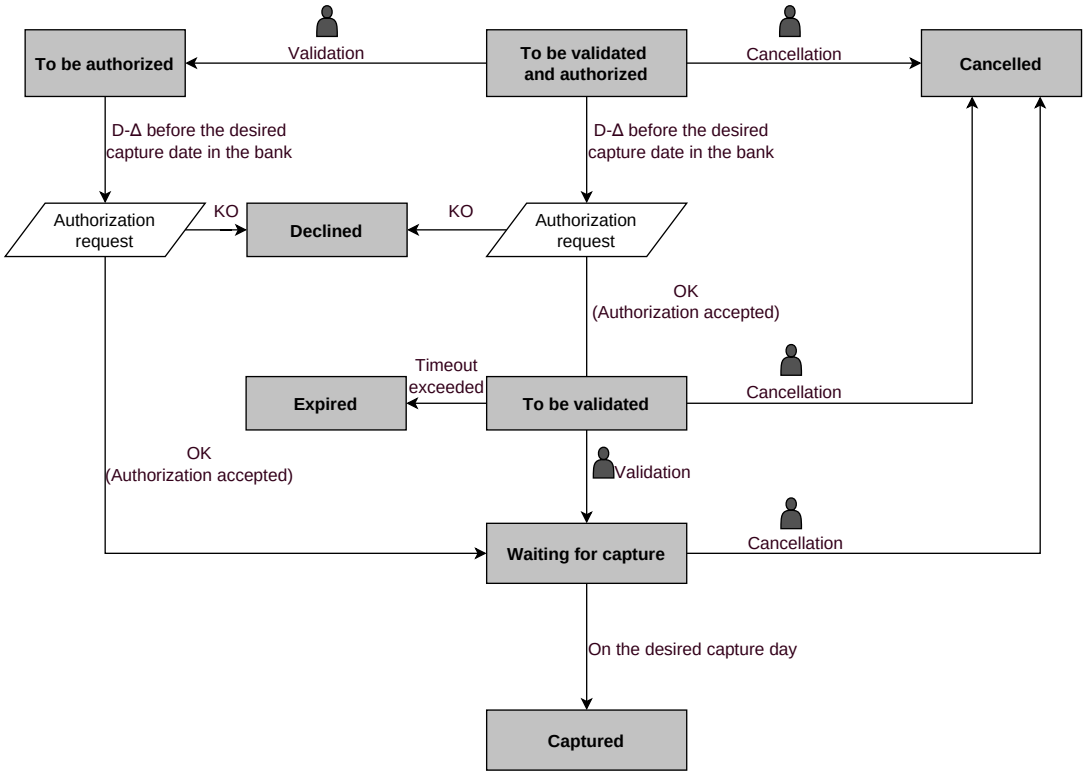
7.3.2. Manual validation

The activation of the payment in installments feature is subject to the prior agreement of Société Générale.

Depending on the capture date, the first installment will have exactly the same features as an immediate payment or a deferred payment.

By default, the upcoming installments have the **To be validated and authorized** status as long as the first installment has not been validated by the merchant. The successful execution of the installments is not guaranteed to the merchant. The buyer's bank may reject the authorization request.

Validation of the first installment implies that all the other installments will be validated as well. However, canceling an installment does not cancel the upcoming installments.



8. ESTABLISHING INTERACTION WITH THE PAYMENT GATEWAY

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 called IPN or Instant Payment Notification), see chapter **Setting up notifications**.
- Via the browser, when the buyer clicks the button to return to the merchant website, see chapter **Managing the 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.

8.1. Setting up the payment page URL

The merchant website interacts with the payment gateway by redirecting the buyer to the following URL:

https://sogecommerce.societegenerale.eu/vads-payment/

8.2. Identifying yourself when exchanging with the payment gateway

To be able to interact with the payment gateway, the merchant needs to have:

- **The shop ID:** allows to identify the merchant website during the exchange. Its value is transmitted in the **vads_site_id** field.
- **The key:** allows to compute the alphanumeric signature transmitted in the **signature** field.

To retrieve these values:

1. Sign in to the Merchant Back Office: *https://sogecommerce.societegenerale.eu/vads-merchant/*
2. Enter your username.

   Your connection identifiers (username and password) are sent to you in an e-mail with the subject **Connection identifiers - [your shop name]**.
3. Enter your password.

   Your connection identifiers (username and password) are sent to you in an e-mail with the subject **Connection identifiers - [your shop name]**.
4. Click **Sign in**.

   The user account is blocked after 3 wrong password entries. If your account is blocked, click **Forgotten password or locked account** to reset it.

   The user password is valid for 90 days. After this period, the user must modify it by logging into their account.
5. Click **Settings > Shop**.
6. Select **Keys**.

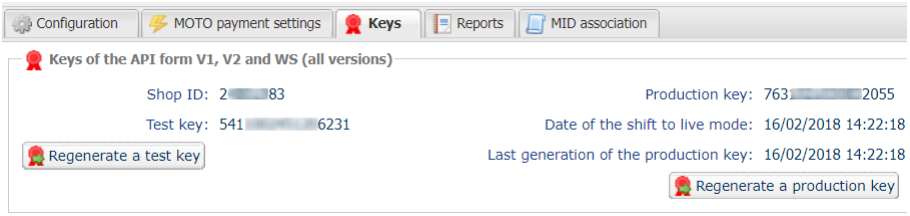


Figure 6: Keys tab

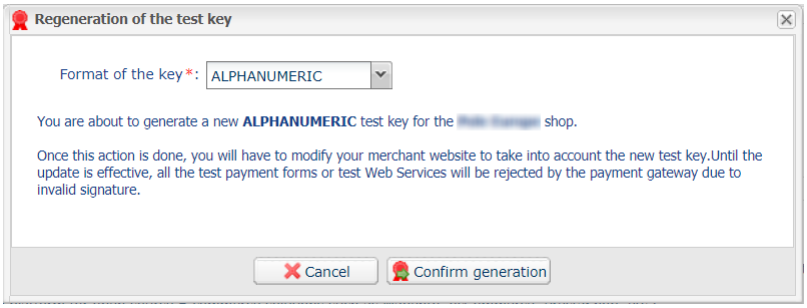
Two types of keys are available:

- The **test key** that allows to generate the form signature in test mode.
- The **production key** that allows you to generate the form signature in production mode.

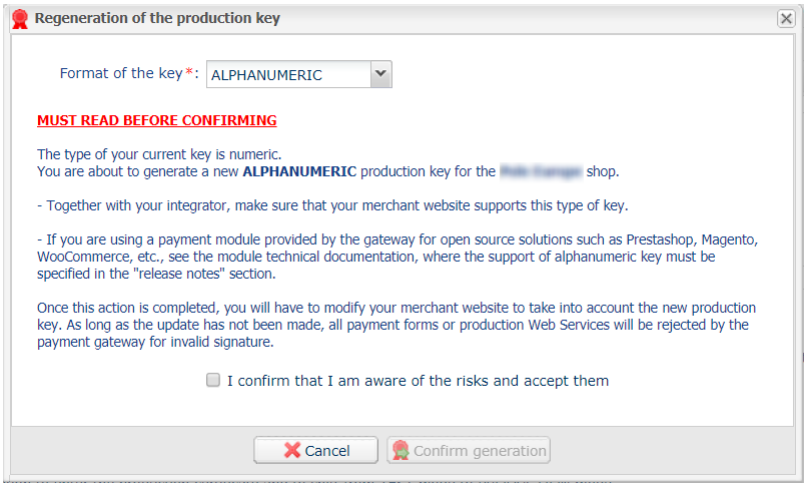
These keys can be numeric or alphanumeric.

For maximum security, it is recommended to use an alphanumeric key.

To change the format of your test key, click the **Regenerate a test key** button and select the format ("ALPHANUMERIC" or "NUMERIC").



To change the format of your production key, click the **Regenerate a production key** button and select the format ("ALPHANUMERIC" or "NUMERIC").



8.3. Choosing between Test and Production modes

The choice between **TEST** or **PRODUCTION** modes can be made using the **vads_ctx_mode** field (See chapter *Generating a payment form* on page 60).

- The **TEST** mode allows to make test payments.

  It is available at all times, even after the generation of the production key.

If you create a new merchant website (or have access to the acceptance testing environment), you can make tests without impacting the website that is currently in production.

TEST transactions can be viewed in the Merchant Back Office via **Management** > **TEST transactions**.

- The **PRODUCTION** mode will become available only once the production key has been generated.

  It allows to make real payments.

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

8.4. Managing interaction with the merchant website

Two types of URLs are used to manage the dialog with the merchant website:

- **Instant Payment Notification**, also called the IPN,
- **URL of return** to the merchant website.

Instant Payment Notification - IPN

The **Notification URL** is the URL of a specific page on the merchant website that is **automatically** called by the payment gateway when certain events take place.

By default, the rules are created to manage the events below:

- end of payment (accepted or rejected),
- payment abandoned or canceled,
- token creation or update,
- recurring payment creation,
- new installment date,
- authorization made in case of a deferred payment,
- update of a transaction status by the acquirer,
- operation made via the Merchant Back Office (cancellation, refund, duplication, manual payment, etc.).

These rules must be enabled and configured according to the needs of the merchant.

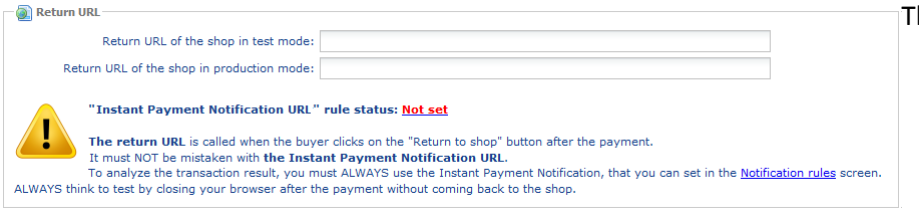
With each call, the payment gateway transmits transaction details to the merchant website. It is called instant notification (or **IPN** as in Instant Payment Notification).

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

URL of return to the merchant website

In the Merchant Back Office, the merchant can configure the "default" return URLs via the menu **Settings** > **Shop** > **Configuration** tab:

Figure 7: Setting up return URLs



The merchant can set up a different return URL for each mode.

By default, the buyer is redirected to the URL regardless of the payment result.

If no URL has been set up, the main URL of the shop will be used for redirection (**URL** parameter defined in the **Details** section of the shop).

The merchant will be able to override this setting in his/her payment form (see chapter **Setting up return URLs**).

The status of the "Instant Payment Notification at the End of Payment" (IPN) rule is displayed in this window. If the URL has not been set up, make sure to specify it (see chapter **Setting up notifications**).

8.5. Managing security

There are several ways to guarantee the security of online payments.

8.5.1. Ensuring interaction integrity

The integrity of exchanged information is preserved by the exchange of alphanumeric signatures between the payment platform and the merchant website.

The payment gateway and the merchant website interact via HTML forms.

A form contains a list of specific fields (see chapter **Generating a payment form**) used to generate a chain.

This chain is then converted to a smaller chain using a hash function (SHA-1, HMAC-SHA-256).

*The merchant will be able to choose the hash algorithm in their Merchant Back Office (see chapter **Choosing the hash algorithm**).*

The resulting chain is referred to as the **digest** (*empreinte* in French) of the initial chain.

The digest must be transmitted in the **signature** field (see chapter **Computing the signature**).

Modeling security mechanisms:

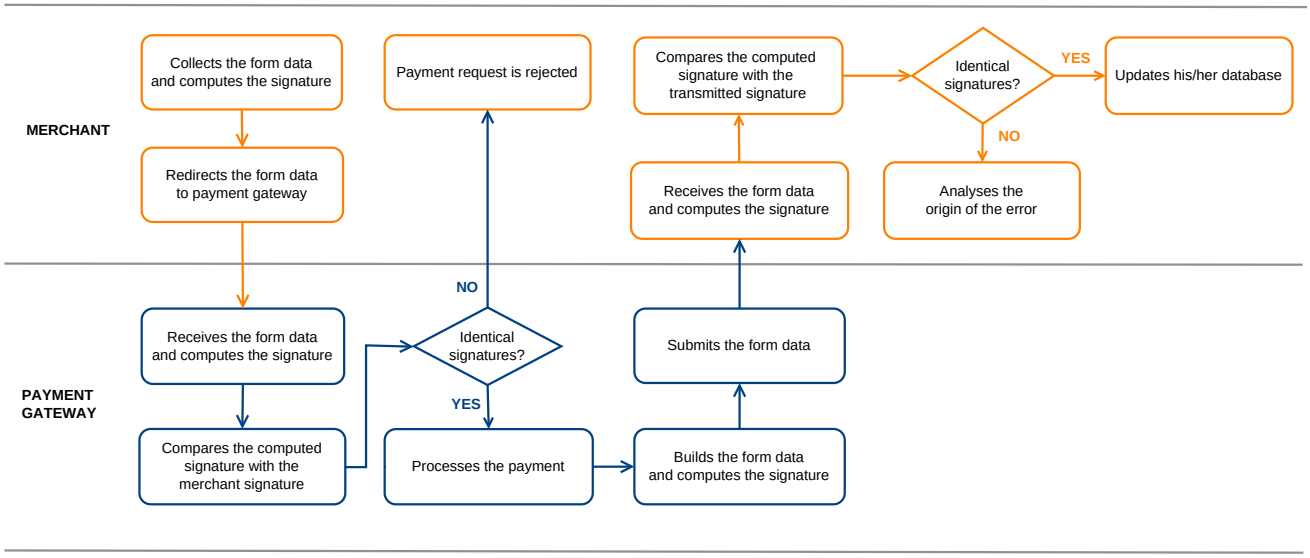


Figure 8: Diagram of a security mechanism

1. The merchant website builds the form data and computes the signature.
2. The merchant website submits the form to the gateway.
3. The gateway receives the form data and computes the signature.
4. The gateway compares the computed signature with the signature that was transmitted by the merchant website.
5. If the signatures are different, the payment request is rejected.

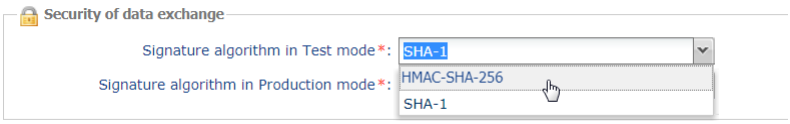
   If not, the gateway proceeds to payment.
6. The gateway builds the result data and computes the response signature.

7. Depending on the shop configuration (see chapter **Setting up notifications**), the payment gateway transmits the payment result to the merchant website.
8. The merchant website receives the data and computes the signature. It compares the computed signature with the signature that was transmitted by the payment gateway.
9. If the signatures are different, the merchant analyses the source of the error (computation error, attempted fraud, etc.).

   If not, the merchant proceeds to update their database (stock status, order status, etc.).

8.5.2. Selecting the hash algorithm

In the Merchant Back Office (**Settings** > **Shop** > **Keys**), the merchant can choose the hash function to use for generating signatures.



HMAC-SHA-256 signature algorithm is applied by default.

You can select a different signature algorithm for TEST mode and for PRODUCTION mode.

However, be sure to use the same method to generate your payment forms and to analyze the data transmitted by the gateway during notifications.

In order to facilitate changing the algorithm, the SHA-1 or HMAC-SHA-256 signatures will be accepted without generating rejections due to signature error for 24h.

8.5.3. Storing the production key

For security reasons, the production key will be masked after the first real payment made with a real card.

It is strongly recommended to store the key in a safe place (encrypted file, database etc.).

In case of losing the key, the merchant will be able to regenerate a new one via their Merchant Back Office.

Remember that the production key can be viewed in the Merchant Back Office via **Settings** > **Shop** > **Keys** tab.

8.5.4. Managing sensitive data

Online payment transactions are regulated by strict rules (PCI-DSS certification).

As a merchant, you have to make sure to never openly transcribe data that could resemble a credit card number. Your form will be rejected (code 999 - Sensitive data detected).

Special attention should be paid to order numbers containing between 13 and 16 numeric characters and beginning with 3, 4 or 5.

8.6. Managing shop settings via a configuration file

Using a configuration file allows to avoid including hard-coded values in the code.

The configuration files may contain:

- the payment page URL,
- the test and production keys,
- the shop ID,
- etc.

These files allow to sort the data to be saved.

The program that generates the payment form interrogates the configuration file to know the value of a parameter.

It is the merchant's responsibility to do anything in his or her power to limit the access to the configuration file (.htaccess file, rewrite the URL, etc.).

Example of "conf.txt" configuration file:

```
vads_site_id = 11111111
TEST_key = 2222222222222222
PROD_key = 3333333333333333
vads_ctx_mode = TEST
```

Example of a call to configuration file in the payment form:

```
$conf_txt = parse_ini_file("conf.txt");
 if ($conf_txt['vads_ctx_mode'] == "TEST") $conf_txt['key'] = $conf_txt['TEST_key'];
 if ($conf_txt['vads_ctx_mode'] == "PRODUCTION") $conf_txt['key'] = $conf_txt['PROD_key'];
```


9. SETTING UP NOTIFICATIONS

The Merchant Back Office allows to manage the events that will generate a notification to the merchant website and to configure the URL of the contact page.

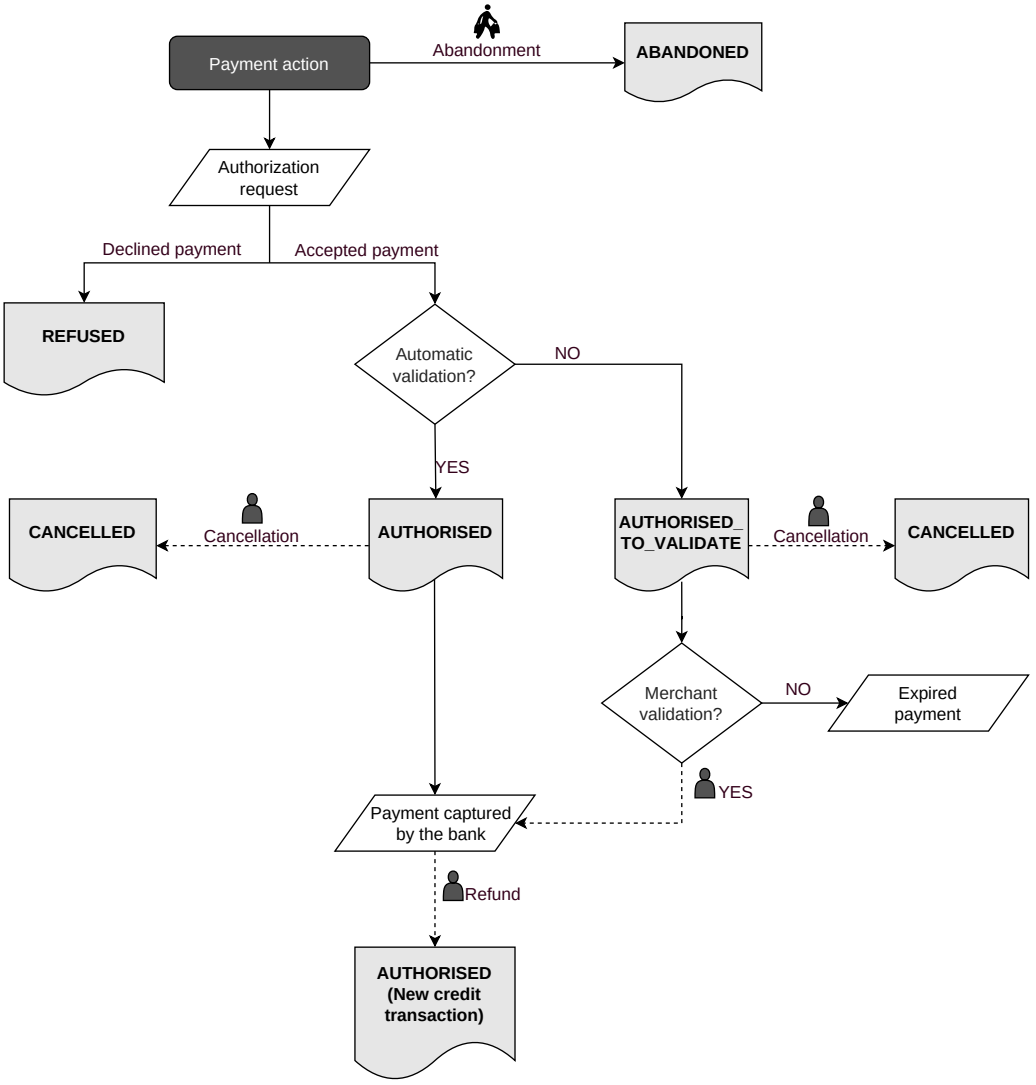
The following diagrams illustrate the transaction status sent in the notification for each event.

The following caption is used for each event:

Action required from the merchant - manual (Merchant Back Office) or automatic (via Web Services)

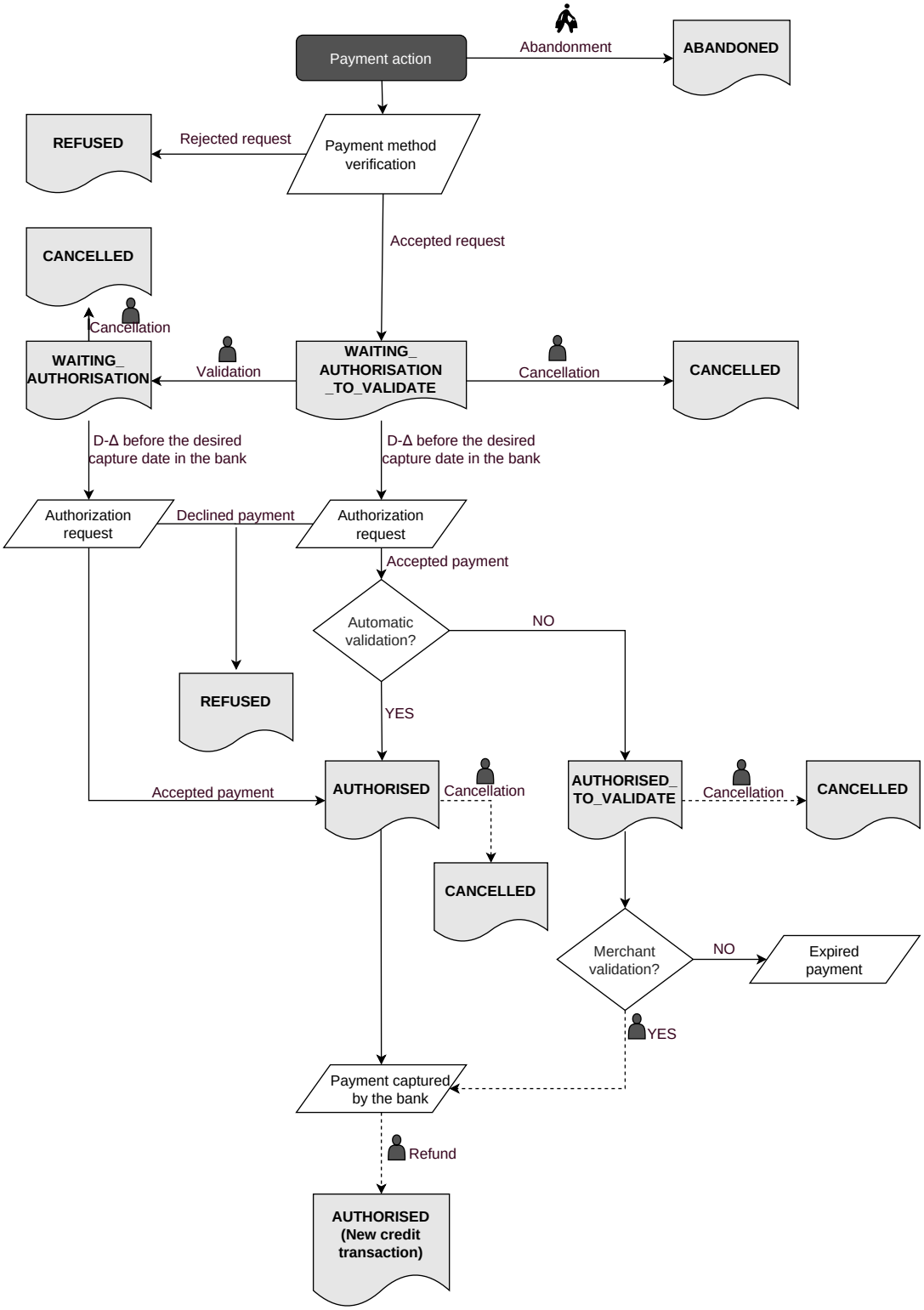
Action performed by the buyer

9.1. Notifications about the various statuses of an immediate payment



Event	Notified status	Name of the rule to configure
Abandoned by the buyer	ABANDONED	Instant Payment Notification URL on cancellation
Cancellation by the merchant	CANCELLED	Instant Payment Notification URL on an operation coming from the Back Office
Response to the authorization request	AUTHORISED_TO_VALIDATE, AUTHORISED, REFUSED	Instant Payment Notification URL at the end of the payment

9.2. Notifications about the different statuses of a deferred payment



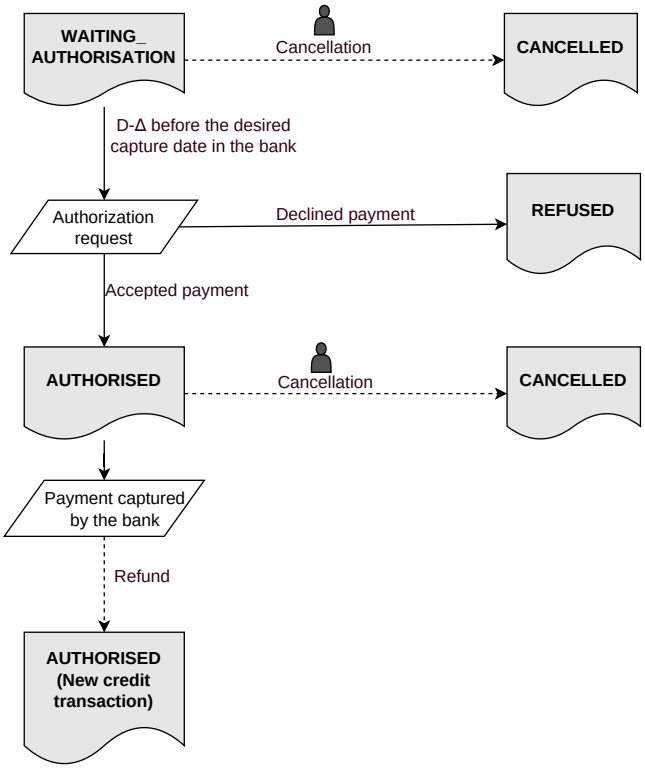
Δ : authorization validity period

Event	Notified status	Name of the rule to configure
Abandoned by the buyer	ABANDONED	Instant Payment Notification URL on cancellation
Cancellation by the merchant	CANCELLED	Instant Payment Notification URL on an operation coming from the Back Office

Event	Notified status	Name of the rule to configure
Cancellation by the merchant	WAITING_AUTHORISATION	Instant Payment Notification URL on an operation coming from the Back Office
Response to the authorization request for EUR 1 (or information request about the CB network if the acquirer supports it)	REFUSED, WAITING_AUTHORISATION, WAITING_AUTHORISATION_TO_VALIDATE	Instant Payment Notification URL at the end of the payment
Response to the authorization request	AUTHORISED, REFUSED, AUTHORISED_TO_VALIDATE	Instant Payment Notification URL on batch authorization

9.3. Notifications about the various statuses of installments

The activation of the payment in installments feature is subject to the prior agreement of Société Générale.



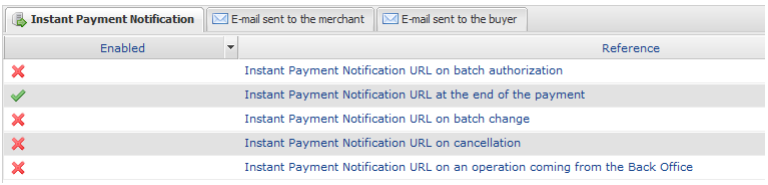
Δ: authorization validity period

Event	Notified status	Name of the rule to configure
Cancellation by the merchant	CANCELLED	Instant Payment Notification URL on an operation coming from the Back Office
Response to the authorization request	AUTHORISED, REFUSED	Instant Payment Notification URL on batch authorization

9.4. Accessing the notification center

Open the **Settings** > **Notification rules** menu.

The rule configuration tab of "Instant Payment Notification URL call" type is displayed.



9.5. Setting up the Instant Payment Notification

The payment gateway notifies on the merchant website in the following cases:

- Payment accepted
- Payment refused
- Token creation or update
- Creation of a recurring payment

The **Payment accepted** event corresponds to the creation of a transaction in one of the (vads_trans_status) statuses below:

- **ACCEPTED**
- **AUTHORISED**
- **AUTHORISED_TO_VALIDATE**
- **CAPTURED**
- **INITIAL**
- **UNDER_VERIFICATION**
- **WAITING_AUTHORISATION**
- **WAITING_AUTHORISATION_TO_VALIDATE**
- **WAITING_FOR_PAYMENT**

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

It informs the merchant website of the payment result even if your client has not clicked the **Return to the shop** button.

1. Right-click **Instant Payment Notification URL at the end of the payment**.
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. To allow the platform to automatically resend the notification in the event of failure, check the **Automatic retry in case of failure** box. This mechanism allows up to 4 attempts to be made.

   For more information, see *Automatic retry in case of failure* on page 57.

5. If you wish to receive notifications in API Form format, enter the URL of your page in **URL to notify in TEST mode** and **URL to notify in PRODUCTION mode** field in the "Instant Payment Notification URL of the API form V1, V2" section.

6. If you're using the clientJavaScript, 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** in the "REST API Instant Payment Notification URL" section.

7. Save the changes.

9.6. Setting up the notification for the final result of a deferred payment

This notification is required for communicating the result of a deferred payment:

- If the payment has been accepted,
- If the payment has been refused.

It allows the merchant website to be notified when the authorization request is not made on the payment day.

Example:

For a deferred payment with a capture delay of 60 days, the authorization request is not made at the moment of the payment. The merchant website will be contacted at the moment of the authorization request by the **Instant Payment Notification URL on batch authorization** rule.

This rule is **disabled by default**.

1. Right-click **Instant Payment Notification URL on batch authorization**.
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. To allow the platform to automatically resend the notification in the event of failure, check the **Automatic retry in case of failure** box. This mechanism allows up to 4 attempts to be made.
   For more information, see *Automatic retry in case of failure* on page 57.
5. If you wish to receive notifications in API Form format, enter the URL of your page in **URL to notify in TEST mode** and **URL to notify in PRODUCTION mode** field in the "Instant Payment Notification URL of the API form V1, V2" section.
6. If you're using the clientJavaScript, 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** in the "REST API Instant Payment Notification URL" section.
7. Save the changes.
8. Enable the rule by right-clicking **Instant Payment Notification URL on batch authorization** and select **Enable the rule**.

9.7. Setting up notifications in case of abandoned or canceled payments

The payment gateway notifies on the merchant website in the following cases:

- When the buyer abandons/cancels a payment - via the **Cancel and return to shop** button.
- When the buyer has not completed the payment and the payment session has expired.

  The maximum length of a payment session is 10 minutes.

This rule is **disabled by default**.

1. Right-click **Instant Payment Notification URL on cancellation**.
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. To allow the platform to automatically resend the notification in the event of failure, check the **Automatic retry in case of failure** box. This mechanism allows up to 4 attempts to be made.

   For more information, see *Automatic retry in case of failure* on page 57.
5. If you wish to receive notifications in API Form format, enter the URL of your page in **URL to notify in TEST mode** and **URL to notify in PRODUCTION mode** field in the "Instant Payment Notification URL of the API form V1, V2" section.
6. If you're using the clientJavaScript, 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** in the "REST API Instant Payment Notification URL" section.
7. Save the changes.
8. Enable the rule by right-clicking **Instant Payment Notification URL on cancellation** and select **Enable the rule**.

9.8. Instant Payment Notification URL on an operation originating from the Back Office

This rule allows to notify the merchant website about every operation made via the Merchant Back Office:

- Creation of a manual payment (accepted or rejected)
- Transaction update
- Transaction duplication
- Transaction refund
- Transaction cancellation
- Transaction validation
- Token creation
- Token update

1. Right-click **Instant Payment Notification URL on an operation coming from the Back Office**.
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).
   For more information, please see chapter *Automatic retry in case of failure* on page 57.
5. If you wish to receive notifications in API Form format, enter the URL of your page in **URL to notify in TEST mode** and **URL to notify in PRODUCTION mode** field in the "Instant Payment Notification URL of the API form V1, V2" section.
6. If you're using the clientJavaScript, 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** in the "REST API Instant Payment Notification URL" section.
7. Save the changes.
8. Enable the rule by right-clicking **Instant Payment Notification URL on an operation coming from the Back Office** and select **Enable the rule**.

9.9. Setting up a notification on batch change

The payment gateway notifies on the merchant website in the following cases:

- When a transaction expires.

  This is the case of transactions created in manual validation mode and that have not been validated in time by the merchant. The status of these transactions changes to "Expired" (EXPIRED).

- When a **PayPal** transaction that has been blocked due to suspected fraud is finally accepted or refused.

  The status of the concerned transactions changes from "Control in progress" (UNDER_VERIFICATION) to "Captured" (CAPTURED) or "Refused" (REFUSED).

- When a **Franfinance** transaction is accepted or refused.

This rule is **disabled by default**.

1. Right-click **Instant Payment Notification URL on batch change**.
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. To allow the platform to automatically resend the notification in the event of failure, check the **Automatic retry in case of failure** box. This mechanism allows up to 4 attempts to be made.

   For more information, see *Automatic retry in case of failure* on page 57.
5. If you wish to receive notifications in API Form format, enter the URL of your page in **URL to notify in TEST mode** and **URL to notify in PRODUCTION mode** field in the "Instant Payment Notification URL of the API form V1, V2" section.
6. If you're using the clientJavaScript, 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** in the "REST API Instant Payment Notification URL" section.
7. Save the changes.
8. Enable the rule by right-clicking **Instant Payment Notification URL on batch change** and select **Enable the rule**.

9.10. Automatic retry in case of failure

Automatic retry does not apply to notifications manually triggered via the Merchant Back Office.

The merchant can enable a mechanism that allows the payment gateway to automatically return notifications when the merchant website is temporarily unavailable. This mechanism allows up to 4 attempts to be made.

A notification will be considered as failed if the HTTP code returned by the merchant site is not on the following list: **200**, **201**, **202**, **203**, **204**, **205**, **206**, **301**, **302**, **303**, **307**, **308**.

Call attempts are scheduled at fixed intervals every 15 minutes (00, 15, 30, 45).

After each failed attempt, a notification e-mail is sent to the e-mail address specified in the configuration of the notification rule in question.

In this case, the subject of the e-mail contains the number corresponding to the notification retry attempt. It is presented as attempt # followed by the attempt number.

- Example of an e-mail subject following a first notification failure at the end of payment:

```
[MODE TEST] My Shop - Tr. ref. 067925 / FAILURE during the call to your IPN URL
 [unsuccessful attempt #1]
```

- Example of an e-mail subject following a second failure:

```
[MODE TEST] My Shop - Tr. ref. 067925 / FAILURE during the call to your IPN URL
 [unsuccessful attempt #2]
```

- Example of an e-mail subject following a third failure:

```
[MODE TEST] My Shop - Tr. ref. 067925 / FAILURE during the call to your IPN URL
 [unsuccessful attempt #3]
```

- Example of an e-mail subject following the last failure:

```
[MODE TEST] My Shop - Tr. ref. 067925 / FAILURE during the call to your IPN URL
 [unsuccessful attempt #last]
```

  To notify the merchant website of the last notification attempt, the e-mail subject will contain the mention attempt #last.

During the automatic retry, certain details are not stored in the database or are modified.

Examples of fields not available/not registered in the database:

Field name	Description
vads_page_action	Completed operation
vads_payment_config	Payment type (immediate or installment).
vads_action_mode	Acquisition mode for payment method data.

Examples of fields sent with different values:

Field name	New value
vads_url_check_src	Always set to **RETRY** in case of automatic retry.
vads_trans_status	The transaction status may vary between the initial call and the automatic retry (cancellation by the merchant, transaction capture at the bank, etc.).
vads_hash	The value of this field is regenerated with each call.
signature	The signature value depends on the different statuses that may vary between the initial call and the automatic retry.

These e-mails contain:

- The encountered problem;
- Parts of analysis depending on the error;
- Its consequences;
- Instructions for manually triggering the notification from the Merchant Back Office.

After the fourth attempt, it is still possible to retry the IPN URL **manually** via your Merchant Back Office.

Note that during the automatic retry, any manual call to the IPN URL will affect the number of automatic attempts:

- A successful manual call will stop the automatic retry;
- A failed manual call will have no impact on the current automatic retry.

9.11. Configuring e-mails sent to the merchant

In the **E-mail sent to the merchant** tab:

1. Right-click the rule to be modified and select **Enable the rule**.
2. Right-click the rule again and select **Manage the rule**.
   The rule management wizard appears.
3. Customize the label of the rule and the address to notify in the General settings section.
   To specify several e-mail addresses, separate them with a semi-colon.
4. In order to customize the body of the e-mail.
   a. Go to **E-mail settings**.
   b. Select the template of the e-mail to apply
   c. Click **Customize default text values** if you wish to edit the body and the subject of the “default” e-mail message.
   d. Click on **Fields to include** to display the list of fields available for e-mail customization.
   e. Select the fields that you wish to include. A detailed summary of the request processing will be added to the body of the e-mail.

   To preview the changes, click **Preview the e-mail** at the bottom of the dialog box.

5. In order to change the events that trigger the notification:
   a. Click the **Rule conditions** tab.
      A condition is composed of a variable, a comparison operator and a reference value.
      Example: "mode = TEST", "amount exceeding 1000". During the execution of a rule, the value of a variable is retrieved and compared to the reference value.
   b. Double-click on an existing condition to edit it.
   c. Click **Add** to create a new condition.

All the conditions must be validated for the rule to be executed.

6. Click **Save**.

9.12. Configuring e-mails sent to the buyer

From the **E-mail sent to the buyer** tab:

1. Right-click the rule to be modified and select **Enable the rule**.
2. Right-click the rule again and select **Manage the rule**.
   The rule management wizard appears.
3. In the General settings section, you can customize the label of the rule.
4. To customize the e-mail content:
   a. Click **Buyer e-mail settings**.
   b. Select the template of the e-mail to apply
   c. Select the language that you would like to update
   d. Click **Customize default text values** if you wish to edit the body and the subject of the “default” e-mail message.
   e. Click on **Fields to include** to display the list of fields available for e-mail customization.
   f. Select the fields that you wish to include. A detailed summary of the request processing will be added to the body of the e-mail.

   
   To preview the changes, click **Preview the e-mail** at the bottom of the dialog box.
5. In order to change the events that trigger the notification:
   a. Click the **Rule conditions** tab.
      A condition is composed of a variable, a comparison operator and a reference value.
      Example: "mode = TEST", "amount exceeding 1000". During the execution of a rule, the value of a variable is retrieved and compared to the reference value.
   b. Double-click on an existing condition to edit it.
   c. Click **Add** to create a new condition.
      All the conditions must be validated for the rule to be executed.
6. Click **Save**.

10. GENERATING A PAYMENT FORM

You must build an HTML form as follows:

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/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 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://sogecommerce.societegenerale.eu/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="parametre1" value="valeur1"/>

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

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

  This will allow for the special characters (accents, punctuation marks, etc.) to be correctly interpreted by the payment gateway. Otherwise, the signature will be computed incorrectly and the form will be rejected.

- The **Pay** button for submitting the data

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

The use cases presented in the following chapters will enable you to build your payment form according to your needs.

Indications on the different possible formats 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.

Notation	Description
	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 key name in quotation marks followed by a colon, followed by a value in quotation marks `"name" : "value"`. The name of the key must be alphanumeric. The value can be: • a string of characters (in this case it must be framed by straight quotes "); • a number; • an object; • a table; • a boolean; • empty. Example: `{"name1":45,"name2":"value2", "name3":false}`
bool	Boolean. Can take the value `true` or `false`.
enum	Defines 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_available_languages=fr;en`
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: • 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`

10.1. Creating an immediate payment

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	Format	Value
vads_site_id	Shop ID	n8	E.g.: 12345678
vads_ctx_mode	Mode of interaction with the payment gateway	enum	TEST or PRODUCTION
vads_trans_id	Transaction number. Must be unique within the same day (from 00:00:00 UTC to 23:59:59 UTC). **Warning: this field is not case sensitive.**	an6	E.g.: xrT15p
vads_trans_date	Date and time of the payment form in UTC format	n14	Respect the YYYYMMDDHHMMSS format E.g.: 20200101130025
vads_amount	Payment amount in the smallest currency unit (cents for euro)	n..12	E.g.: 4525 for EUR 45.25
vads_currency	Numeric currency code to be used for the payment, in compliance with the ISO 4217 standard (numeric code).	n3	E.g.: 978 for euro (EUR)
vads_action_mode	Acquisition mode for payment method data	enum	INTERACTIVE
vads_page_action	Action to perform	enum	PAYMENT
vads_version	Version of the exchange protocol with the payment gateway	enum	**V2**
vads_payment_config	Payment type	enum	SINGLE
vads_payment_cards	Allows to force the card type to be used. It is recommended to provide a different payment button for each payment method on the merchant website. **It is recommended not to leave the field empty**. See the chapter *Managing the payment methods offered to the buyer* on page 90 for more information.	enum	E.g.: • CB • CVCONNECT • MASTERCARD • VISA • SDD
vads_capture_delay	Capture delay	n..3	
vads_validation_mode	Validation mode	n1	**0** (Automatic)
signature	Signature guaranteeing the integrity of the requests exchanged between the merchant website and the payment gateway.	ans..44	Compute the value of the signature field using all the fields of your form starting with **vads_** (see chapter ***Computing the signature***).

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

3. Set the **vads_capture_delay** field to **0**.

4. Set the **vads_validation_mode** field to **0** for automatic validation (the payment will be automatically captured in the bank).

5. Populate the **vads_currency** field with the code of the desired currency using the *currency table* (E.g.: 978 for euro (EUR)).

6. Add ***the fields recommended for increasing chances of frictionless*** during the payment.

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

Example of a form for an immediate payment:

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_amount" value="15000" />
<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_order_id" value="CX-1254" />
<input type="hidden" name="vads_page_action" value="PAYMENT" />
<input type="hidden" name="vads_payment_cards" value="CB" />
<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="pt156G" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="0WaYrONo3L0VZqMcvyVf8vT/g8KfZKJ+1jqiAs3Ehiw="/>
<input type="submit" name="pay" value="Pay"/>
</form>
```


10.2. Creating a deferred payment

A deferred payment is a payment debited all at once with a capture delay that is strictly greater than 0 days.

An information request will be made if the capture delay is greater than the validity period of an authorization request (see chapter *Authorization request validity period* on page 22).

The information request is made in order to check the card validity. For acquirers who do not support information requests, an authorization request for EUR 1 will be made.

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

Field name	Description	Format	Value
vads_site_id	Shop ID	n8	E.g.: 12345678
vads_ctx_mode	Mode of interaction with the payment gateway	enum	TEST or PRODUCTION
vads_trans_id	Transaction number. Must be unique within the same day (from 00:00:00 UTC to 23:59:59 UTC). **Warning: this field is not case sensitive.**	an6	E.g.: xrT15p
vads_trans_date	Date and time of the payment form in UTC format	n14	Respect the YYYYMMDDHHMMSS format E.g.: 20200101130025
vads_amount	Payment amount in the smallest currency unit (cents for euro)	n..12	E.g.: 4525 for EUR 45.25
vads_currency	Numeric currency code to be used for the payment, in compliance with the ISO 4217 standard (numeric code).	n3	E.g.: 978 for euro (EUR)
vads_action_mode	Acquisition mode for payment method data	enum	INTERACTIVE
vads_page_action	Action to perform	enum	PAYMENT
vads_version	Version of the exchange protocol with the payment gateway	enum	**V2**
vads_payment_config	Payment type	enum	SINGLE
vads_payment_cards	Allows to force the card type to be used. It is recommended to provide a different payment button for each payment method on the merchant website. **It is recommended not to leave the field empty**. See the chapter *Managing the payment methods offered to the buyer* on page 90 for more information.	enum	E.g.: • CB • CVCONNECT • MASTERCARD • VISA • SDD
vads_capture_delay	Delay before capture in the bank, **the value must be greater than 0**	n..3	E.g.: 3
vads_validation_mode	Specifies the validation mode of the transaction (manually by the merchant, or automatically by the payment gateway).	n1	**0** or **1** or **absent** or **empty**
signature	Signature guaranteeing the integrity of the requests exchanged between the merchant website and the payment gateway.	ans..44	Compute the value of the signature field using all the fields of your form starting with **vads_** (see chapter ***Computing the signature***).

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

3. Set the **vads_capture_delay** field to a value **greater than 0**.

4. Set the **vads_validation_mode** field to **0** for an automatic validation (the payment will be automatically captured at the bank) or to **1** for a manual validation (the payment will be captured in the bank after a manual validation in the Merchant Back Office).

5. Fill in the **vads_currency** field with the code of the desired currency using the *currency table* (E.g.: 978 for euro (EUR)

6. Add ***the fields recommended for increasing chances of frictionless*** during the payment.

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

Example of a form for a deferred payment:

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/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="3" />
<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_cards" value="CB" />
<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="20190629130025" />
<input type="hidden" name="vads_trans_id" value="Hu92ZQ" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="NrHSHyBBBc+TtcauudspNHQ5cYcy4tS4IjvdC0ztFe8="/>
<input type="submit" name="pay" value="Pay"/>
</form>
```


10.3. Creating an installment payment

The activation of the payment in installments feature is subject to the prior agreement of Société Générale.

Under PSD2, strong authentication is required upon the payment of the first installment. The vads_threeds_mpi field is ignored and the CHALLENGE_MANDATE value is automatically applied.

This payment mode allows the merchant to offer payment facilities to the buyer.

The payment form defines the number of installments and the interval between them.

The first installment works the same way as an immediate payment.

The next installment(s) is similar to (a) deferred payment(s).

Reminder:

Notification rules have to be activated depending on the installment. See chapter ***Setting up notifications*** for more information.

Details:

The **vads_amount** field contains the total amount of the order. This is the amount that will be split according to the value of the **vads_payment_config** field.

On the payment day, the total amount is not credited to the merchant's account and the payment guarantee cannot apply to future installments.

The date of the last installment cannot exceed one year after the date of the form submission. Otherwise, an error message will appear and the form will be rejected.

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

Field name	Description	Format	Value
vads_site_id	Shop ID	n8	E.g.: 12345678
vads_ctx_mode	Mode of interaction with the payment gateway	enum	TEST or PRODUCTION
vads_trans_id	Transaction number. Must be unique within the same day (from 00:00:00 UTC to 23:59:59 UTC). **Warning: this field is not case sensitive.**	an6	E.g.: xrT15p
vads_trans_date	Date and time of the payment form in UTC format	n14	Respect the YYYYMMDDHHMMSS format E.g.: 20200101130025
vads_amount	Payment amount in the smallest currency unit (cents for euro)	n..12	E.g.: 4525 for EUR 45.25
vads_currency	Numeric currency code to be used for the payment, in compliance with the ISO 4217 standard (numeric code).	n3	E.g.: 978 for euro (EUR)
vads_action_mode	Acquisition mode for payment method data	enum	INTERACTIVE
vads_page_action	Action to perform	enum	PAYMENT
vads_version	Version of the exchange protocol with the payment gateway	enum	**V2**
vads_payment_config	Payment type	enum	See step 2.
vads_payment_cards	Allows to force the card type to be used.	enum	E.g.: • CB

Field name	Description	Format	Value
	It is recommended to provide a different payment button for each payment method on the merchant website. **It is recommended not to leave the field empty**. See the chapter *Managing the payment methods offered to the buyer* on page 90 for more information.		• MASTERCARD • VISA
vads_capture_delay	Capture delay	n..3	
vads_validation_mode	Specifies the validation mode of the transaction (manually by the merchant, or automatically by the payment gateway).	n1	**0** or **1** or **absent** or **empty**
signature	Signature guaranteeing the integrity of the requests exchanged between the merchant website and the payment gateway.	ans..44	Compute the value of the signature field using all the fields of your form starting with **vads_** (see chapter ***Computing the signature***).

2. Populate the **vads_payment_config** field using the following syntax:

   - Fixed payment amounts and dates:

     MULTI:first=1000;count=3;period=30 where:

     "first" corresponds to the amount (in the smallest currency unit) of the first installment made on the day of payment,

     "count" represents the total number of installments,

     "period" determines the interval between each installment.

   - Custom installment amounts and dates:

     MULTI_EXT:date1=amount1;date2=amount2;date3=amount3 where:

     date1=amount1 defines the date and the amount of the first transfer.

     The amounts are presented in the smallest currency unit. The total amount must be equal to the value of the **vads_amount** field.

     The dates are presented in the YYYYMMDD format.

3. Set the **vads_capture_delay** field to **0**. The first payment will be captured in the bank on the same day.

4. Set the **vads_validation_mode** field to **0** for automatic validation (the payment will be automatically captured in the bank) or to **1** for manual validation (manual operation performed via the Merchant Back Office).

   The validation mode applies to all the installments.

5. Populate the **vads_currency** field with the code of the desired currency using the *currency table* (E.g.: 978 for euro (EUR)).

6. Add ***the fields recommended for increasing chances of frictionless*** during the payment.

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

Example of installment payment form (fixed amounts and payment dates):

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_amount" value="3000" />
<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="MULTI:first=1000;count=3;period=30"/>
<input type="hidden" name="vads_site_id" value="12345678" />
<input type="hidden" name="vads_trans_date" value="20190629180150" />
<input type="hidden" name="vads_trans_id" value="1N015m" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value= "zrhUNkAciZSEl6mS4BbhV3qkYUBB9RYJQCdg1kU0ELU="/>
<input type="submit" name="pay" value="Pay" />
</form>
```

Example of installment payment form (custom amounts and payment dates):

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/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="
MULTI_EXT:20140201=1000;20140301=1000;20140401=1000" />
<input type="hidden" name="vads_site_id" value="12345678" />
<input type="hidden" name="vads_trans_date" value="20190629130025" />
<input type="hidden" name="vads_trans_id" value="130025" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="7Sds6Z+R1Q1axRsblpChyQh5OU3oCle5FOirD4V/Bzk="/>
<input type="submit" name="pay" value="Pay"/>
</form>
```


10.4. Creating an authorization without capture

This payment mode allows to make sure that the buyer's card data is correct without debiting it.

If needed, the merchant will be able to debit the desired amount from the card account by using the **Duplicate** function of the **Merchant Back Office.** To do this:

- the manual validation mode is used,
- the merchant must not validate transactions manually.

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

Field name	Description	Format	Value
vads_site_id	Shop ID	n8	E.g.: 12345678
vads_ctx_mode	Mode of interaction with the payment gateway	enum	TEST or PRODUCTION
vads_trans_id	Transaction number. Must be unique within the same day (from 00:00:00 UTC to 23:59:59 UTC). **Warning: this field is not case sensitive.**	an6	E.g.: xrT15p
vads_trans_date	Date and time of the payment form in UTC format	n14	Respect the YYYYMMDDHHMMSS format E.g.: 20200101130025
vads_amount	Payment amount in the smallest currency unit (cents for euro)	n..12	E.g.: 4525 for EUR 45.25
vads_currency	Numeric currency code to be used for the payment, in compliance with the ISO 4217 standard (numeric code).	n3	E.g.: 978 for euro (EUR)
vads_action_mode	Acquisition mode for payment method data	enum	INTERACTIVE
vads_page_action	Action to perform	enum	PAYMENT
vads_version	Version of the exchange protocol with the payment gateway	enum	**V2**
vads_payment_config	Payment type	enum	SINGLE
vads_capture_delay	Capture delay	n..3	
vads_validation_mode	Validation mode	n1	**1** (Manual)
signature	Signature guaranteeing the integrity of the requests exchanged between the merchant website and the payment gateway.	ans..44	Compute the value of the signature field using all the fields of your form starting with **vads_** (see chapter ***Computing the signature***).

2. Set the value of the **vads_amount** field to a small amount. It will not affect the authorization limit of the card.

3. Set the **vads_validation_mode** field to **1**.

4. Populate the **vads_currency** field with the code of the desired currency using the *currency table* (E.g.: 978 for euro (EUR)).

5. Add optional fields according to your requirements (see chapter **Using additional features**).

Example of a form for an authorization without capture:

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_amount" value="100" />
<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_validation_mode" value="1"/>
<input type="hidden" name="vads_site_id" value="12345678" />
<input type="hidden" name="vads_trans_date" value="20190628073753" />
<input type="hidden" name="vads_trans_id" value="3jj7A8" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value= "DvltInRYXRroOZ/KnNdJSlpVr++29ZGty4nj1Y7yczU="/>
<input type="submit" name="pay" value="Pay" />
</form>
```


11. USING ADDITIONAL FEATURES

11.1. Managing the return to the merchant website

At the end of payment, the buyer has the possibility to return to the merchant website via a return URL. This URL is called **Return URL**.

It is not to be confused with **Instant notification URL (IPN)** (see chapter ***Managing the interaction with the merchant website***).

11.1.1. Defining the Return URLs

In the payment form, the merchant can override the configuration of the Merchant Back Office. To do so, the merchant can:

- Use 4 different URLs depending on the payment result:
  - Payment accepted
  - Payment declined
  - Payment abandoned
  - Payment error
- Or use a single URL independently of the payment result.

Defining the return URLs depending on the payment result

Use the optional fields presented in the table below to create a customized payment form.

If no URL is specified in the form, the value populated in the Merchant Back Office will be used.

Field name	Description	Format	Value
vads_url_cancel	URL to which the buyer will be redirected upon clicking on “Cancel and return to shop” before proceeding to the payment.	ans..1024	E.g.: http://demo.com/cancel.php
vads_url_error	URL to which the buyer will be redirected in case of a processing error on the payment gateway.	ans..1024	E.g.: http://demo.com/error.php
vads_url_refused	URL to which the buyer will be redirected in case of a declined payment after having clicked on “Return to shop”.	ans..1024	E.g.: http://demo.com/refused.php
vads_url_success	URL to which the buyer will be redirected in case of an accepted payment after having clicked on “Return to shop”.	ans..1024	E.g.: http://demo.com/success.php

Example of a payment form with configuration of a return URL depending on the payment result:

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/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="PRODUCTION" />
<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="20191126101407" />
<input type="hidden" name="vads_trans_id" value="pm197W" />
<input type="hidden" name="vads_url_cancel" value="http://demo.com/cancel.php" />
<input type="hidden" name="vads_url_error" value="http://demo.com/error.php" />
<input type="hidden" name="vads_url_refused" value="http://demo.com/refused.php" />
<input type="hidden" name="vads_url_success" value="http://demo.com/success.php" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="lZIHzigiwCc6+uLStp8I5DQnbSqXu63Jtfo6Saeq3Mc="/>
<input type="submit" name="pay" value="Pay"/>
</form>
```

Setting up a unique return URL regardless of the payment outcome

Use the optional field **vads_url_return** to set up a redirection URL at the end of payment.

If no URL is specified in the form, the value populated in the Merchant Back Office will be used.

Example of a payment form with a unique return URL regardless of the payment result:

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/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="PRODUCTION" />
<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="20191126101407" />
<input type="hidden" name="vads_trans_id" value="xrTYh2" />
<input type="hidden" name="vads_url_return" value="http://demo.com/return.php" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="ZI/GhI0GbeqqoXGeoZuPOy55SKQSYzRO1i6r5ku6vOs="/>
<input type="submit" name="pay" value="Pay"/>
</form>
```


11.1.2. Defining the method for receiving data

For statistical purposes or to display customized pages, the merchant site must be able to analyze certain data transmitted to the buyer's browser.

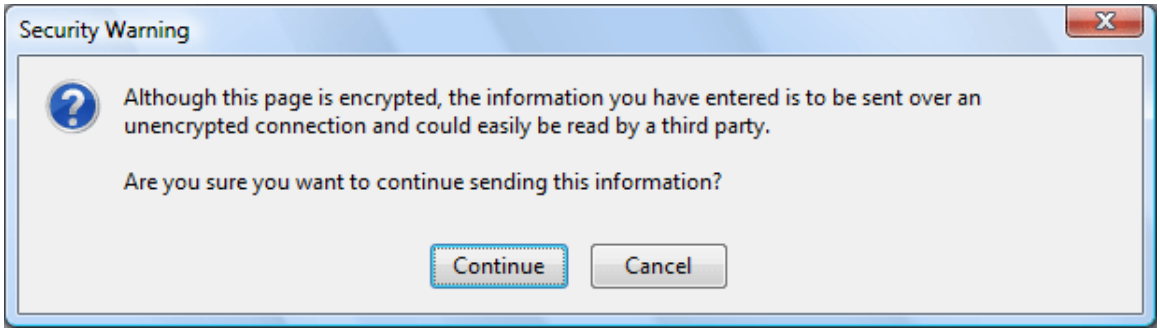
By default, the payment gateway does not transmit any data when redirecting to the return URL.

However, the merchant website may activate the transmission of data to the return URL via the payment form.

Use the **vads_return_mode** optional field to specify the method for submitting data to the merchant website.

Value	Description
Absent, empty or **NONE**	No data is transmitted.
GET	The data is transmitted in the browser URL.
POST	The date is transmitted via an HTTP in **POST** request.

The **GET** method allows to keep a notification message from appearing when the return is done from an **insecure environment (http)**.



Example of a payment form with definition of the mode for data transmission:

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/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="PRODUCTION" />
<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_return_mode" value="GET" />
<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="239848" />
<input type="hidden" name="vads_url_return" value="http://demo.com/return.php" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="oTCT+7Oc+xttdGmcp9qa6/0pSSfNxoMtl8U1J1l+LtE="/>
<input type="submit" name="pay" value="Pay"/>
</form>
```


11.2. Enabling automatic return to the merchant website

In the payment form, the merchant can indicate if he/she wishes to automatically redirect the buyer to the merchant website at the end of payment.

If you use a tracking code (e.g. Google Analytics™) on your website, you must implement this function.

1. Use optional fields according to your requirements.

Field name	Description
vads_redirect_success_timeout	Defines the delay before redirection following an accepted payment. This delay is presented in seconds and must be between 0 and 300 sec.
vads_redirect_success_message	Defines the message that appears before redirection following a successful payment.
vads_redirect_error_timeout	Defines the delay before the redirection that follows a declined payment. This delay is presented in seconds and must be between 0 and 300 sec.
vads_redirect_error_message	Defines the message that appears before the redirection that follows a declined payment.

If you set the timeout to zero (= 0 delay) your redirection will be done as follows:

- *For an **accepted payment**, the buyer will be redirected to **vads_url_success**.*
- *For a **canceled payment**, the buyer will be redirected to **vads_url_cancel** if this parameter is defined.*
  - *If the parameter is not defined, the buyer will be redirected to the return URL entered in the **vads_url_return** field or to the return URL entered in the Merchant Back Office.*
  - *If the return URL is not set, the buyer will be redirected to the merchant website.*
- *For a **declined payment**, the buyer will be redirected to **vads_url_refused** if the setting is defined.*

2. Set the **vads_return_mode** field to **GET**.

Example of a payment form:

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/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="PRODUCTION" />
<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_redirect_error_message" value="You will be redirected to your
 merchant website" />
<input type="hidden" name="vads_redirect_error_timeout" value="0" />
<input type="hidden" name="vads_redirect_success_message" value="You will be redirected to your
 merchant website" />
<input type="hidden" name="vads_redirect_success_timeout" value="0" />
<input type="hidden" name="vads_return_mode" value="GET" />
<input type="hidden" name="vads_site_id" value="AL3d8Q" />
<input type="hidden" name="vads_trans_date" value="20190626101407" />
<input type="hidden" name="vads_trans_id" value="239848" />
<input type="hidden" name="vads_url_return" value="http://demo.com/return.php" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="AzTJmizS5N0muYzu63nVvCUWo0ixnMJfpqQmuEa4CSY="/>
<input type="submit" name="pay" value="Pay"/>
</form>
```


11.3. Defining the capture mode (automatic/manual)

In the Merchant Back Office, the merchant can configure how payments are sent to the bank (**Settings** > **Shop** menu > **Configuration** tab):

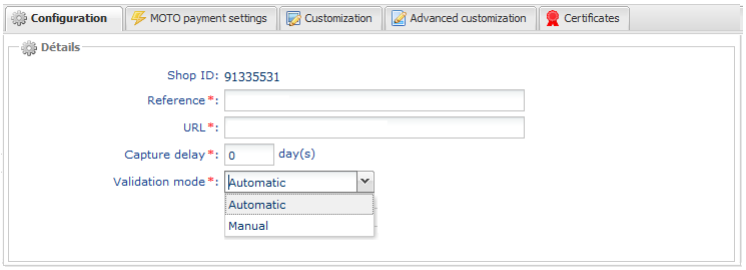


Figure 9: Defining the capture mode

- **Automatic**: no action is necessary, the payments are captured in the bank once the capture delay has been reached.
- **Manual**: the merchant must validate each payment via their Merchant Back Office or Web Services, so that it is captured in the bank before the authorization request expires.

  Each transaction that has not been validated by the expected date is considered as expired and will never be captured in the bank.

  By default, the Merchant Back Office is configured to automatically submit all payments to the bank.

  The merchant can override this configuration in their payment form.

  The merchant must implement the desired criteria (stock status, delay for stock replenishment, etc.) allowing the buyer to decide whether the transaction must be captured automatically or not.

  Use the **vads_validation_mode** field to configure the capture mode (manual or automatic).

  This field will be resent with the response and will include the value transmitted in the form.

Value	Description
Missing or empty	Takes the value defined in the Merchant Back Office.
0	Automatic capture. Transaction is automatically validated by the payment gateway.
1	Manual capture. The transaction must be validated manually by the merchant via their Merchant Back Office (or automatically via the **Transaction/Validate** Web Service function).

Example of a payment form with a definition of the capture mode in INTERACTIVE mode:

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/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="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="20190626164147" />
<input type="hidden" name="vads_trans_id" value="164147" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="vads_validation_mode" value="1" />
<input type="hidden" name="signature" value="cJFhNTLXQ4o6BgbW1pMMoM2yMilw90OIqmFjJ6DeUmA= />
<input type="submit" name="pay" value="Pay"/>
</form>
```


11.4. Transmitting buyer details

The Merchant can specify the buyer details (e-mail address, title/civil status, 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_national_id	National identifier	ans..255	E.g.: 940992310285
vads_cust_title	Buyer's title	an..63	E.g.: M
vads_cust_status	Status	enum	PRIVATE: for private clients **COMPANY**: for companies
vads_cust_first_name	First name	ans..63	E.g.: Laurent
vads_cust_last_name	Last name	ans..63	E.g.: Durant
vads_cust_legal_name	Buyer's legal name	ans..100	E.g.: D. & Cie
vads_cust_phone	Phone number	an..32	E.g.: 0467330222
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	Address line 2	ans..255	E.g.:
vads_cust_district	District	ans..127	E.g.: Centre ville
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 alpha-2 standard	a2	E.g.: "FR" for France, "PF" for French Polynesia, "NC" for New Caledonia, "US" for the United States.

Example of payment form with buyer details:

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/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="smith.john@example.com" />
<input type="hidden" name="vads_cust_first_name" value="John" />
<input type="hidden" name="vads_cust_last_name" value="Smith" />
<input type="hidden" name="vads_cust_title" value="Mr" />
<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="20190627133115" />
<input type="hidden" name="vads_trans_id" value="522754" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="rEFhNTLXQ4o6BgbW1pTMoM2yMilw90OIqmFjJ6DeCxP="/>
<input type="submit" name="pay" value="Pay"/>
</form>
```


11.5. 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 (**Shipping 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.: Bordeaux
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.: 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	Last 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	Address line 2	ans..255	
vads_ship_to_zip	Zip code	an..64	E.g.: 33000

Example of payment form with shipping details:

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/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_page_action" value="PAYMENT" />
<input type="hidden" name="vads_payment_config" value="SINGLE" />
<input type="hidden" name="vads_ship_to_city" value=" city of the shipping address" />
<input type="hidden" name="vads_ship_to_country" value="FR" />
<input type="hidden" name="vads_ship_to_name" value="location name of the shipping address" />
<input type="hidden" name="vads_ship_to_street" value="street of the shipping address" />
<input type="hidden" name="vads_ship_to_street_number" value="10" />
<input type="hidden" name="vads_ship_to_zip" value="31670" />
<input type="hidden" name="vads_site_id" value="12345678" />
<input type="hidden" name="vads_trans_date" value="20190627143509" />
<input type="hidden" name="vads_trans_id" value="561095" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="bOIxHAgm4vYUq3oIDCdEPKOWgrB9bHzkfDBEAr1i10A="/>
<input type="submit" name="pay" value="Pay"/>
</form>
```


11.6. 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).

1. Use optional fields according to your requirements.

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	ans..255	E.g.: Door code 3125
vads_order_info2	Additional order info	ans..255	E.g.: No elevator
vads_order_info3	Additional order info	ans..255	E.g.: Express
vads_nb_products	Number of items in the cart	n..12	E.g.: 2
vads_product_ext_idN	Product barcode on the merchant website. N corresponds to the index of the item (0 for the first one, 1 for the second one, etc.).	an..100	E.g.: vads_product_ext_id0 = "0123654789123654789" vads_product_ext_id1 = "0223654789123654789" vads_product_ext_id2 = "0323654789123654789"
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	Price of the item incl. VAT. 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	Item quantity. 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"

Field name	Description	Format	Value
			vads_product_qty2 = "2"

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

> 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 Merchant 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_**.*

3. 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.).

4. 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 / Gardening
HOME_APPLIANCE	Household appliances
AUCTION_AND_GROUP_BUYING	Auctions / Group purchasing
FLOWERS_AND_GIFTS	Flowers / Presents
COMPUTER_AND_SOFTWARE	Computers / Software
HEALTH_AND_BEAUTY	Health / Beauty
SERVICE_FOR_INDIVIDUAL	Services for individuals
SERVICE_FOR_BUSINESS	Services for companies
SPORTS	Sports
CLOTHING_AND_ACCESSORIES	Clothes / Accessories
TRAVEL	Travel
HOME_AUDIO_PHOTO_VIDEO	Audio / Photo / Video
TELEPHONY	Telephony

5. 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.).

6. 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.).

7. 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.).

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

Example of the payment form with "vads_product_xxx" cart description:

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_amount" value="11000" />
<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_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_trans_date" value="20190627145218" />
<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="payer" value="Payer"/></form>
```


11.7. Transmitting merchant preferences

Use the **vads_threeds_mpi** field to transmit your preferences:

Use case	Values	Description
CHALLENGE: with cardholder interaction	**1**	**Deprecated**.
	3	**3DS Requestor Preference**: Allows to request strong authentication for the transaction.
	4	**Challenge request mandate**: Allows to indicate that, due to regulatory reasons, strong authentication is required for the transaction.
FRICTIONLESS: without cardholder interaction	**2***	Allows to Request an exemption from strong authentication: • Low value transactions. • Transactional Risk Analysis (TRA Acquéreur). • LRM (Low Risk Merchant). More informations: Table of exemptions, below.
No merchant preference	**0 or absent or empty**	The choice of the preference is transferred to the card issuer. If the issuer decides to perform an authentication without interaction (frictionless), the payment will be guaranteed.
	5	

***Table of exemptions** (value number 2):

Exemptions	Description
Low value transactions	In Europe, you can request an exemption from strong authentication, for transactions of less than €30, and within the limit of either 5 successive operations or a cumulative amount of less than €100. If the amount is higher than €30, the value transmitted by the merchant is ignored and the choice of the preference is transferred to the card issuer (No Preference). For payments made in a currency other than euro, a request for frictionless is transmitted to the issuer. If the frictionless request is accepted, the transaction does not benefit from liability shift dispute by the cardholder.. If the store does not have the “Frictionless 3DS2” option, the choice of the preference is transferred to the card issuer (No Preference).
Transactional Risk Analysis (TRA Acquéreur)	If your store has the "TRA Acquirer 3DS2" option, you can ask the issuer for an exemption from strong authentication if the amount is below the threshold set by your financial institution. If the frictionless request is accepted, the transaction does not benefit from liability shift dispute by the cardholder.. The “Acquirer 3DS2 TRA" activation option is subject to the prior agreement of your financial institution.

LRM (Low Risk Merchant)	CB offers the LRM (=Low Risk Merchant) program. This program is designed to meet the needs of very low-risk, high-volume merchants. You can request an exemption from strong authentication:
	• If the amount is less than €100, the exemption is systematic for eligible merchants.
	• If the amount is between €100 and €250, an experiment is underway. To qualify, the merchant must :
	  • Have a CB contract.
	  • Be eligible for TRA acquéreur.
	  • Transmit the required values in the 3D Secure flow, according to the rules defined by the platform.
	If the frictionless request is accepted, the transaction does not benefit from liability shift dispute by the cardholder..

Exemptions	Description
	To benefit from CB's LRM program, you must contact your customer advisorSociété Générale to obtain explicit approval.

11.8. Overriding the Instant Payment Notification (IPN) URL

You can override the Instant Payment Notification (also called IPN) in the payment form in case you use one shop for various sales channels, payment types, languages, etc.

This feature is not compatible with the execution of the request sent to the IPN via the Merchant Back Office. The called URL is the URL that was set up in the notification rule (see chapter ***Setting up notifications***).

Use the **vads_url_check** field to override the URL of the page to notify.

If the value of the **vads_url_check** field is wrong, the form will be rejected.

Example of payment form with IPN override:

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/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="PRODUCTION" />
<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="20190626101407" />
<input type="hidden" name="vads_trans_id" value="239848" />
<input type="hidden" name="vads_url_check" value="http://www.myshop.com/check" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="yXvSZnYvcMRORVGiapWaHT0euKDI0OGlrddYKc4XDZc="/>
<input type="submit" name="pay" value="Pay"/>
</form>
```


11.9. Defining the Merchant ID (MID)

In the payment form, the Merchant must specify the value of the Merchant ID (MID).

This feature is used only if you have several MIDs that accept the same currency within the same acceptance network.

Use the optional **vads_contracts** field to define the Merchant ID (MID) to be used.

- To **define a list** of MIDs, separate them with a semi-colon ";".

```
vads_contracts=NETWORK_CODE_A=MID_A1;NETWORK_CODE_B=MID_B2
```

- To **exclude a network**, add **network name**=**NO**.

```
vads_contracts=NETWORK_CODE_A=NO
```

- To **force the TID**, separate the MID number and the TID number by a colon: ": "

```
vads_contracts=NETWORK_CODE_A=MID_A1:TID_1
```

- If the field is submitted empty, the MID used will be the one defined by the priority order in the Merchant Back Office (**Settings** > **Shop** > **MID association** tab).

List of available networks:

Network code	Description
ACCORD_SANDBOX	Oney network (private and gift cards) - sandbox mode
ACCORD	Oney network (private and gift cards)
AMEXGLOBAL	American Express network
AURORE	Cetelem Aurore network (Brand cards and universal Aurore card)
CB	CB network
CONECS	Titre-Restaurant Conecs network
FULLCB	FULL CB Network (Payment in 3 or 4 times without fees by BNPP PF)
MASTERPASS	MasterPass network
PAYCONIQ	Payconiq network
PAYPAL	PayPal network
PAYPAL_SB	PayPal network - sandbox mode
SEPA	SEPA network (SDD and SCT)

Examples:

You have:

- Two MIDs within the A network: MID_A1 and MID_A2
- Two MIDs within the B network: MID_B1 and MID_B2

To specify which MID to use for these two networks, **vads_contracts** must be populated as follows:

```
vads_contracts=A=MID_A2;B=MID_B1
```

To offer a payment only for the MID_A1 contract and prevent payments within the B network, populate **vads_contracts** as follows:

```
vads_contracts=A=MID_A1;B=NO
```


In order to force the TID to be used within the A network:

```
vads_contracts=A=MID_A1:TID_A1
```


11.10. Creating specific fields according to your requirements

The merchant can transmit specific information in the payment form. For example, the merchant can add information in the payment confirmation e-mail that he or she will receive.

This information will be visible in the Back Office, in transaction details (**Extras** tab), and will also be returned in the notification URL.

The name must begin with **vads_ext_info** to be taken into account.

vads_ext_info_*fieldname*=value

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

2. Use the optional field **vads_ext_info** depending on your needs and respecting the syntax:

   vads_ext_info_*fieldname*=*value*

   Where:

   - ***fieldname***

     Allows to define the name of the field.

   - ***value***

     Allows to define the value of the field.

   There are no restrictions to the number of specific fields that can be created.

   This/these field(s) will be returned with the response and will include the value transmitted in the form.

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://sogecommerce.societegenerale.eu/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="TEST" />
<input type="hidden" name="vads_currency" value="978" />

[...]
<input type="hidden" name="vads_ext_info_qty_articles" value="2" /> />
[...]
<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="20150826133115" />
<input type="hidden" name="vads_trans_id" value="722754" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="7896adcaf7338930db9715afa123531f42"/>
<input type="submit" name="pay" value="Pay"/>
</form>
```


11.11. Transmitting sub-merchant details

The payment facilitator can transmit the details of the sub-merchant involved in the transaction.

This information can be found in the Merchant Back Office by viewing the transaction details (**Sub-merchant** tab).

Field name	Description	Format
vads_submerchant_address	Address of the sub-merchant. Transmitted by the payment facilitator.	ans..255
vads_submerchant_address2	Address line 2 of the sub-merchant. Transmitted by the payment facilitator.	ans..255
vads_submerchant_city	City of the sub-merchant. Transmitted by the payment facilitator.	ans..128
vads_submerchant_company_type	Company type of the sub-merchant. Transmitted by the payment facilitator.	ans..60
vads_submerchant_country	Country of the sub-merchant's address (ISO 3166 alpha-2 standard). Transmitted by the payment facilitator.	a2
vads_submerchant_facilitatorId	Payment Facilitator ID. Transmitted by the payment facilitator.	ans..128
vads_submerchant_legal_number	Legal Entity Identifier of the sub-merchant. Transmitted by the payment facilitator.	ans..24
vads_submerchant_mcc	Merchant Category Code of the sub-merchant. Transmitted by the payment facilitator.	n4
vads_submerchant_mid	Merchant ID number of the sub-merchant. Transmitted by the payment facilitator.	n..64
vads_submerchant_name	Legal name of the sub-merchant. Transmitted by the payment facilitator.	ans..255
vads_submerchant_phone	Phone number of the sub-merchant. Transmitted by the payment facilitator.	ans..32
vads_submerchant_soft_descriptor	Soft descriptor of the sub-merchant that appears on the buyer's bank statement. Transmitted by the payment facilitator.	ans..255
vads_submerchant_state	Region of the sub-merchant address. Transmitted by the payment facilitator.	ans..128
vads_submerchant_url	URL of the sub-merchant. Transmitted by the payment facilitator.	ans..128
vads_submerchant_zip	Zip code of the sub-merchant. Transmitted by the payment facilitator.	an..64

12. CUSTOMIZING ELEMENTS ON THE PAYMENT PAGE

Allows to customize certain elements on the payment page:

- the payment methods offered at the moment of payment,
- the language used for displaying the payment pages,
- the languages offered to the buyer on the payment pages,
- the name and the URL of the shop,
- button labels.

Thanks to the **advanced customization** option, you can also:

- create different custom templates of the payment page in order to make it look more similar to your merchant website,
- create different custom templates of e-mails sent to the buyer,
- customize certain labels that appear on the payment pages.

This will result in a better user experience during redirection to proceed to payment.

Consult the *Advanced customization* user manual for more details or contact your customer advisorSociété Générale.

12.1. Overriding the custom template

The customization feature of the payment pages is subject to prior agreement of Société Générale.

The Merchant Back Office allows:

- to create several custom templates of payment pages,
- to define the template that will be used by default for all your transactions.

The payment form allows to dynamically override the template to be used thanks to the **vads_theme_config** field.

For this, you must use the keyword: **RESPONSIVE_MODEL** and indicate the name of the template to be used (Model_1, Model_2, ...).

Example of use:

```
<input type="hidden" name="vads_theme_config" value="RESPONSIVE_MODEL=Model_1" />
```

See the *Back Office user manual - Advanced customization* for more details on template creation.

See the ***vads_theme_config*** chapter for more details on using this field.

12.2. Managing the payment methods offered to the buyer

It is possible to customize the payment methods offered to the buyer by using the **vads_payment_cards** field.

It is recommended to provide a different payment button for each payment method on the merchant website and to transmit the buyer's choice in the **vads_payment_cards** field.

The list of possible values is available in the *Data dictionary*.

For more information, please consult the documentation dedicated to each payment method that you wish to offer.

To offer payment by CB, Visa, Mastercard, Maestro, Visa Electron and e-CB cards, we recommend to only submit the **CB** value.

To offer payment by card via European acquirers (Elavon, Six, Concardis, VR Pay, etc.), we recommend to submit the "**VISA**" or "**MASTERCARD**" value.

Thus, the buyer is redirected to the card data entry page, and the card type is automatically detected.

It is strongly recommended not to leave this field empty. In case of adding a new payment method to your shop, it will be offered automatically, even if you do not wish to offer it.

Example of a payment form with payment method selection:

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/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="CB" />
<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="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>
```


12.3. Selecting a different language

You can customize the language of the payment pages.

Populate **vads_language** with one the values presented in the table below.

Language	ISO 639-1 standard
German	de
English	en
Chinese	zh
Spanish	es
French	fr
Italian	it
Japanese	ja
Dutch	nl
Polish	pl
Portuguese	pt
Russian	ru
Swedish	sv
Turkish	tr

- If the value of the **vads_language** field is wrong, the form will be rejected.
- If the field has not been sent or is empty, the payment page will be shown in the language of the buyer's browser.
- The buyer will be able to change the language anytime by using the language selector in the top right corner of the payment page.

Example of a payment form with a list of available languages:

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/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="PRODUCTION" />
<input type="hidden" name="vads_currency" value="978" />
<input type="hidden" name="vads_language" value="fr" />
<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="239848" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="PAMdHJ8FJc2CqUJLXQLxz+e77K4k1YGJmI5mHqGN74g="/>
<input type="submit" name="pay" value="Pay"/>
</form>
```


12.4. Modifying the languages available to the buyer

You can customize the list of languages offered to the buyer using the language selector at the top right of the payment page.

The last language selected by the buyer will be the default language for the payment confirmation e-mail.

Populate the **vads_available_languages** using the table below:

- with one single value, if you do not wish to show the page of payment method selection,
- **with a list of values separated by a ";"** to show the available languages.

Language	Value	Default available language
German	de	x
English	en	x
Chinese	zh	x
Spanish	es	x
French	fr	x
Italian	it	x
Japanese	ja	x
Dutch	nl	x
Polish	pl	
Portuguese	pt	x
Russian	ru	x
Swedish	sv	x
Turkish	tr	

If the value of the **vads_available_languages** field is wrong, the form will be rejected.

Example of a payment form with a list of available languages:

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_amount" value="3000" />
<input type="hidden" name="vads_available_languages" value="fr;en;nl;de" />
<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_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="239848" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="aEWutqzuHH6Q8ns3a6cj5XitZCuhYsDcsKjlLpL8flA="/>
<input type="submit" name="pay" value="Pay"/>
</form>
```


12.5. Modifying the name and the URL of the shop

If you have two domain names, you can modify the name and the URL of the shop to make the domain name visible.

1. Use the **vads_shop_name** field to override the name of the shop that appears on the summary payment page, the receipt and the confirmation e-mail.

2. Use the **vads_shop_url** field to modify the shop URL that appears on the payment pages.

   This value will be used for the confirmation e-mail.

   If the value of the **vads_shop_url** field is wrong, the form will not be rejected.

Example of a payment form including the modification of the shop name and URL:

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/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="PRODUCTION" />
<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_shop_name" value="My Shop" />
<input type="hidden" name="vads_shop_url" value="http://www.myshop.com" />
<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="239848" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="gV0f2HZzQ9BxttHM2W5ZM+AKQsxu0HjDvKy0NAE/G24="/>
<input type="submit" name="pay" value="Pay"/>
</form>
```


12.6. Changing the name of the "Return to shop" button

You can customize the text of the button **"Return to the shop"**.

1. Use the **vads_theme_config** field to change the name of the "Return to shop" button.

2. Use the **SUCCESS_FOOTER_MSG_RETURN** keyword to change the name of the "Return to shop" button that appears if the payment has been accepted.

3. Use the **CANCEL_FOOTER_MSG_RETURN** keyword to change the name of the "Cancel and return to shop" button that appears on payment pages.

By subscribing to the **advanced customization** option, you will be able to change the names (e.g.: shop ID) of the buttons on the payment page.

See: Back Office user manual *Advanced customization* for more details or contact your customer advisorSociété Générale.

Example of payment form with modification of the name of the "Return to shop" button:

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/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_order_id" value="CD100000858" />
<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_theme_config"
 value="CANCEL_FOOTER_MSG_RETURN=Cancel;SUCCESS_FOOTER_MSG_RETURN=Return" />
<input type="hidden" name="vads_trans_date" value="20190631092024" />
<input type="hidden" name="vads_trans_id" value="408248" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="ge5DHBbUGsq4cFfSIR1QyB/L/9qPNp2vhX9/G3kKJeQ="/>
<input type="submit" name="pay" value="Pay"/>
</form>
```


13. 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 key value 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.

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://sogecommerce.societegenerale.eu/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=

13.1. Example of implementation with JAVA

Definition of the utility class SHA that will include the elements required to process the HMAC-SHA-256 algorithm

```
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import java.io.UnsupportedEncodingException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.util.Base64;
import java.util.TreeMap;

public class VadsSignatureExample {
 /**
  * Build signature (HMAC SHA-256 version) from provided parameters and secret key.
  * Parameters are provided as a TreeMap (with sorted keys).
  */
 public static String buildSignature(TreeMap<String, String> formParameters, String
  secretKey) throws NoSuchAlgorithmException, InvalidKeyException, UnsupportedEncodingException
 {
   // Build message from parameters
   String message = String.join("+", formParameters.values());
   message += "+" + secretKey;
   // Sign
   return hmacSha256Base64(message, secretKey);
  }
    /**
    * Actual signing operation.
    */
 public static String hmacSha256Base64(String message, String secretKey) throws
  NoSuchAlgorithmException, InvalidKeyException, UnsupportedEncodingException {
   // Prepare hmac sha256 cipher algorithm with provided secretKey
   Mac hmacSha256;
   try {
    hmacSha256 = Mac.getInstance("HmacSHA256");
   } catch (NoSuchAlgorithmException nsae) {
    hmacSha256 = Mac.getInstance("HMAC-SHA-256");
   }
   SecretKeySpec secretKeySpec = new SecretKeySpec(secretKey.getBytes("UTF-8"), "HmacSHA256");
   hmacSha256.init(secretKeySpec);
   // Build and return signature
   return Base64.getEncoder().encodeToString(hmacSha256.doFinal(message.getBytes("UTF-8")));
  }
}
```

Definition of the utility class SHA that will include the elements required for processing the SHA-1 algorithm

```
import java.security.MessageDigest;
import java.security.SecureRandom;

public class Sha {
    static public final String SEPARATOR = "+" ;
    public static String encode(String src) {
        try {
            MessageDigest md;
            md = MessageDigest.getInstance( "SHA-1" );
            byte bytes[] = src.getBytes( "UTF-8" );
            md.update(bytes, 0, bytes. length );
            byte[] sha1hash = md.digest();
            return convertToHex(sha1hash);
        }
        catch(Exception e){
            throw new RuntimeException(e);
        }
    }
    private static String convertToHex(byte[] sha1hash) {
        StringBuilder builder = new StringBuilder();
            for (int i = 0; i < sha1hash. length ; i++) {
               byte c = sha1hash[i];
               addHex(builder, (c >> 4) & 0xf);
               addHex(builder, c & 0xf);
        }
        return builder.toString();
    }
    private static void addHex(StringBuilder builder, int c) {
        if (c < 10)
            builder.append((char) (c + '0' ));
        else
            builder.append((char) (c + 'a' - 10));
```



```
        }
}
```


Function that computes the signature:

```
public ActionForward performCheck(ActionMapping actionMapping, Basivoirorm form,
    HttpServletRequest request, HttpServletResponse response){
        SortedSet<String> vadsFields = new TreeSet<String>();
        Enumeration<String> paramNames = request.getParameterNames();

        // retrieve and sort the fields starting with vads_* alphabetically
        while (paramNames.hasMoreElements()) {
            String paramName = paramNames.nextElement();
            if (paramName.startsWith( "vads_" )) {
                vadsFields.add(paramName);
            }
        }
// Compute the signature
String sep = Sha.SEPARATOR;
StringBuilder sb = new StringBuilder();
for (String vadsParamName : vadsFields) {
    String vadsParamValue = request.getParameter(vadsParamName);
    if (vadsParamValue != null) {
        sb.append(vadsParamValue);
    }
    sb.append(sep);
}
sb.append( shaKey );
String c_sign = Sha.encode(sb.toString());
return c_sign;}
```


13.2. Example of implementation with PHP

Example of signature computation using the HMAC-SHA-256 algorithm:

```
function getSignature ($params,$key)
{
    /**
     *Function that computes the signature.
     * $params : table containing the fields to send in the payment form.
     * $key : TEST or PRODUCTION key
     */
    //Initialization of the variable that will contain the string to encrypt
    $signature_content = "";

    //sorting fields alphabetically
    ksort($params);
    foreach($params as $name=>$value){

        //Recovery of vads_ fields
        if (substr($name,0,5)=='vads_'){

            //Concatenation with "+"
            $signature_content .= $value."+";
         }
    }
    //Adding the key at the end
    $signature_content .= $key;

    //Encoding base64 encoded chain with SHA-256 algorithm
    $signature = base64_encode(hash_hmac('sha256',$signature_content, $key, true));
    return $signature;
 }
```

Example of signature computation using the SHA-1 algorithm:

```
function getSignature($params, $key)
{
    /**
     * Function that computes the signature.
     * $params : table containing the fields to send in the payment form.
     * $key : TEST or  PRODUCTION key
     */
    //Initialization of the variable that will contain the string to encrypt
    $signature_content = "" ;

    // Sorting fields alphabetically
    ksort($params);
        foreach ($params as $name =>$value)
    {
    // Recovery of vads_ fields
        if (substr($name,0,5)=='vads_') {
            // Concatenation with "+"
            $signature_content .= $value."+";
        }
    }
    // Adding the key at the end
    $signature_content .= $key;

    // Applying SHA-1 algorithm
    $signature = sha1($signature_content);
    return $signature ;
}
```


14. SENDING THE PAYMENT REQUEST

To finalize a purchase, the buyer must be redirected to the payment page.

His browser must transmit the payment form data.

14.1. Redirecting the buyer to the payment page

The URL of the payment gateway is:

https://sogecommerce.societegenerale.eu/vads-payment/

Example of parameters sent to the payment gateway:

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_amount" value="1315" />
<input type="hidden" name="vads_currency" value="978" />
<input type="hidden" name="vads_cust_id" value="1234" />
<input type="hidden" name="vads_cust_email" value="jg@sample.com" />
<input type="hidden" name="vads_ctx_mode" value="TEST" />
<input type="hidden" name="vads_order_id" value="CMD012859" />
<input type="hidden" name="vads_page_action" value="PAYMENT" />
<input type="hidden" name="vads_payment_cards" value="VISA;MASTERCARD" />
<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="20200326101407" />
<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>
```

14.2. Processing errors

If the payment gateway detects an error while receiving the form, an error message is displayed and the buyer can not be 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 indicates to the buyer that a technical problem has occurred.

In both cases, the merchant receives a warning e-mail containing the information:

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

The e-mail is sent to the company administrator.

To change this address or add an address, contact your customer advisorSociété Générale.

You can also create a personalized notification rule to receive this e-mail at another address.

To do so:

1. Sign in to your Merchant Back Office:

   https://sogecommerce.societegenerale.eu/vads-merchant/
2. Open the **Settings** > **Notification rules** menu.
3. Select **Advanced notification**.
4. Select the type of **E-mail sent to the merchant** notification.
5. Click **Next**.
6. Select the trigger event for **Invalid payment form**.
7. In the **General settings**, fill in the fields:
   - **Rule reference**
   - **E-mail address to notify**
8. Click **Create**.

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

https://sogecommerce.societegenerale.eu/doc/fr-FR/error-code/error-00.html

Other messages may appear during the payment process.

Here is a list of the most frequent messages:

Message	Description
This website does not currently accept payments.	"Seasonal service" is enabled in your shop. Payments are only authorized during the defined period. If you want to modify the opening period of your shop, please contact your customer advisorSociété Générale.
Your payment request has been declined by your financial institution.	• The buyer's bank has rejected the authorization or information request. • The risk assessment rules have triggered the rejection of the transaction.
Your registration request has been declined by your financial institution.	• The buyer's bank has rejected the authorization or information request. • The risk assessment rules have triggered the rejection of the transaction.
This payment order is expired. Please contact your shop.	The buyer clicked on the payment link after the payment order expiration date.
This payment order has already been paid.	The buyer clicked on the payment link one more time after having already made the payment.
An error occurred during the payment request, the merchant website has been informed of the impossibility to finalize the transaction.	The payment form has been rejected. The shop administrator has received an e-mail with the details about the origin of the error.
The transaction has already been made.	The merchant website sends a transaction identifier that has already been used for another transaction (accepted or rejected). The transaction identifier must be unique within the same day (00:00:00 at 23:59:59 UTC).
Sorry, you have been disconnected due to a long period of inactivity.	• The buyer attempts to validate the card number while the payment session is expired. The session is open for about 10 minutes. • The merchant website sends a transaction identifier that has already been used but that did not result in a transaction (e.g. abandoned payment). The transaction

Message	Description
	identifier must be unique within the same day (00:00:00 at 23:59:59 UTC).
Cookies are blocked by your browser. Make sure you authorize them before retrying the operation.	The buyer has disabled cookies in his or her browser. Cookies are necessary for the payment to be processed correctly.

14.3. Managing timeouts

Payment session

A "payment session" is the time spent by a buyer on the payment page.

The payment session begins as soon as the payment gateway receives the payment form.

The delay of payment session is 10 minutes (except for certain payment methods).

This delay is:

- **sufficient** to enable each buyer to make his or her payment
- **fixed in time**: it is not reset after every action of the user
- **non-modifiable**: it is fixed by the payment gateway due to technical constraints

After this delay, the payment session times out and the session data is purged.

Expiration of the payment session

In some cases the payment session will expire while the buyer has not completed the payment.

Most frequent cases:

1. For example, once redirected to the payment page, the buyer realizes that it is time to go to lunch.

   An hour later, the buyer decides to continue his or her payment and clicks on the logo corresponding to his or her payment method.

   The buyer's payment session has already expired, the payment gateway displays an error message indicating that the buyer was disconnected due to an extended period of inactivity.

   The buyer then has the opportunity to click a button to return to the merchant website.

   The return to the shop is done via the URL specified by the merchant:

   - in the *vads_url_return* field transmitted in the payment form,
   - in the "Return URL to the merchant website" field in the buyer's Merchant Back Office, no *vads_url_return* field is transmitted in his or her payment form.
2. Once redirected to the payment page, the buyer closes the browser (by mistake or because he or she no longer wants to make the payment).

Notification in case of session expiration

It is possible to notify the merchant website in case of expiration of the payment session.

To do this, the merchant must set up and activate the ***notification on cancellation*** rule (see chapter *Setting up notifications*).

15. IMPLEMENTING THE IPN

The script must include at least the following steps:

- Retrieve the field list sent with the POST response
- Compute the signature taking into account the received data
- 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 already been updated.

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

In order to facilitate support and diagnosis by the merchant in the event of a notification error, we recommend to write messages that will allow you to know at which stage of processing the error occurred.

The gateway reads and stores the first 256 bytes of the HTTP response.

You can write messages throughout the processing. Here are some examples of messages that you can use:

Message	Use case
Data received.	Message to display when retrieving data. Allows to confirm that the notification has been received by the merchant website.
POST is empty.	Message to display when retrieving data. Allows to bring out a possible redirection that would have caused the parameters posted by the payment gateway to be lost.
An error occurred while computing the signature.	Message to be displayed when the verification of the response signature has failed.
Order successfully updated.	Message to be displayed at the end of the file once your processing has been successfully completed.
An error occurred while updating the order.	Message to be displayed at the end of the file if an error occurred during your processing.

15.1. Preparing your environment

The notifications of Instant Payment Notification URL call type are very important as they represent the only reliable way for the merchant website to obtain the payment result.

It is therefore necessary to make sure the notifications function properly.

Here are some guidelines:

- In order for the dialog between the payment gateway and your merchant website to work, you must make sure, together with your technical teams, that the **194.50.38.0/24** IP address range is authorized on the various devices within your system (firewalls, apache server, proxy server, etc.)

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

- Using redirection leads to losing data presented in POST.

  This is the case if there is a configuration on your devices or on the side of your host that redirects the URLs of “http://**www.**example.com” type to “http://example.com” or “**http**://example.com” to “**https**://example.com”.

- HTML must not be visible on the page. Access to images or CSS slows down the exchange between the payment gateway and the merchant website.

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

  The processing time has a direct influence on the time 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 considers that the call has failed (timeout).

- If your page is only accessible in https, test your URL on the Qualys SSL Labs website (*https://www.ssllabs.com/ssltest/* and, if necessary, change your configuration if necessary in order to obtain the A score.

  Your SSL certificate must be signed by a certification authority known and recognized on the market.

- Make sure that you use the latest version of the TLS protocol in order to maintain a high level of security.

15.2. Retrieving data returned in the response

The data returned in the response depends on the parameters sent in the payment request, the payment type, the settings of your shop and the notification format.

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 response consists of a field list. Each field contains a response value. The field list can be updated.

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

It is recommended to test the presence of the **vads_hash** field, which is only present during a notification.

```
if (empty ($_POST)){
 echo 'POST is empty';

}else{
 echo 'Data Received ';
 if (isset($_POST['vads_hash'])){

  echo 'Form API notification detected';
  //Signature computation
  //Signature verification
  //Order Update
 }
}
```


15.3. Computing the IPN signature

The signature is computed by following the same logic as for creating the payment request.

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 the payment request.

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. **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.

Examples in PHP:

```
function getSignature ($params,$key)
{
    /**
     *Function that computes the signature.
     * $params: table containing the fields received in the IPN.
     * $key : TEST or PRODUCTION key
     */
    //Initialization of the variable that will contain the string to encrypt
    $signature_contents = "";

    //Sorting fields alphabetically
    ksort($params);
    foreach($params as $name=>$value){

        //Recovery of vads_ fields
        if (substr($name,0,5)=='vads_'){

            //Concatenation with "+"
            $signature_contents .= $value."+";
         }
    }
    //Adding the key at the end
    $signature_contents .= $key;

    //Encoding base64 encoded chain with HMAC-SHA-256 algorithm
    $sign = base64_encode(hash_hmac('sha256',$signature_contents, $key, true));
    return $sign;
 }
```


15.4. Comparing signatures

To ensure the integrity of the response, you must compare the signature contained in the IPN with the value computed in the previous step.

You should not compare the signature of the IPN with the signature that you transmitted in your payment request.

If the signatures match

- You may consider the response as safe and proceed with the analysis.
- Otherwise, the script will have to raise an exception and notify the merchant about the anomaly.

Example in PHP:

```
if ($_POST['signature'] == $sign){

    //Processing data

}else{
    throw new Exception('An error occurred while computing the signature');
}
```

The signatures may not match in case of:

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

15.5. Analyzing the nature of the notification

During a 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 Merchant 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 Only if the merchant has configured the **Instant Payment Notification URL at the end of payment** rule. • payment abandoned or canceled by the buyer Only if the merchant has configured the **Instant Payment Notification URL on cancellation** rule.
BO	Execution of the notification via the Merchant 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. This is the case of payments with redirection to the acquirer. 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 **Instant Payment Notification URL on batch authorization** rule. 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 an operation made via the Merchant Back Office (refund, cancellation, modification, validation, duplication, creation and/or update of token), only if the merchant has configured the following notification rule: **Instant Payment Notification URL on an operation coming from the Back Office**
RETRY	Automatic retry of the IPN.

Table 1: 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.

15.6. Processing the response data

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

1. Identify the mode (TEST or PRODUCTION) that was used for creating the transaction by analyzing the value of the **vads_ctx_mode** field.

2. Identify the order by retrieving the value of the **vads_order_id** field if you have transmitted it to the payment gateway.

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

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

   Its value allows you to define the order status.

Value	Description
ABANDONED	**Abandoned** Payment abandoned by the buyer The transaction has not been created, and **therefore cannot be viewed in the Merchant Back Office**.
ACCEPTED	**Accepted**. Status of a VERIFICATION type transaction for which the authorization request or information request has been successfully completed. This status cannot evolve. Transactions with the **Accepted** status will never be captured.
AUTHORISED	**Waiting for capture** The transaction has been accepted and will be automatically captured at the bank on the expected date.
AUTHORISED_TO_VALIDATE	**To be validated** The transaction, created with manual validation, is authorized. The merchant must manually validate the transaction in order for it to be captured. The transaction can be validated as long as the expiration date of the authorization request has not passed. If the authorization validity period has been passed, the payment takes **Expired** status. This status is final.
CANCELLED	**Cancelled** The transaction has been canceled by the Merchant.
CAPTURED	**Captured** The transaction has been captured by the bank.
CAPTURE_FAILED	**Capture failed** Contact the technical support.
EXPIRED	**Expired** This status appears in the lifecycle of a payment with deferred capture. 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.
REFUSED	**Refused** The transaction is refused.
SUSPENDED	**Suspended** The capture of the transaction is temporarily blocked by the acquirer (AMEX GLOBAL or SECURE TRADING). Once the transaction has been correctly captured, its status changes to **CAPTURED**.
UNDER_VERIFICATION	**Control in progress** Waiting for the response from the acquirer. This status is temporary.

Value	Description
	A notification will be sent to the merchant website to inform the Merchant of the status change. Requires the activation of the Instant Payment Notification URL on batch change notification rule.
WAITING_AUTHORISATION	**Waiting for authorization** The capture delay in the bank exceeds the authorization validity period.
WAITING_AUTHORISATION_TO_VALIDATE	**To be validated and authorized** The capture delay in the bank exceeds the authorization validity period. A EUR 1 (or information request about the CB network if the acquirer supports it) authorization has been accepted. The merchant must manually validate the transaction for the authorization request and the capture to occur.

4. Analyze the **vads_occurrence_type** field to determine if it is a single payment or a payment that is part of a series (subscription or installment payment).

Value	Description
UNITAIRE	Single payment (immediate payment).
RECURRENT_INITIAL	First payment of a series.
RECURRENT_INTERMEDIAIRE	Nth payment of a series.
RECURRENT_FINAL	Last payment of a series.

5. Analyze the **vads_payment_config** field to determine whether it is an **installment payment**.

Field name	Value for an immediate payment	Value for a payment in installments
vads_payment_config	SINGLE	MULTI (the exact syntax is MULTI:first=X;count=Y;period=Z)

For a payment in installments, identify the installment number by retrieving the value of the **vads_sequence_number** field.

Warning: with the application of Soft Decline, the **vads_sequence_number** field no longer allows to easily identify the first installment of a payment in installments. Since the sequence number of the first installment can be different from 1, the sequence number of the second installment will not necessarily be 2.

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

7. Retrieve the value of the **vads_capture_delay** field to identify the number of days before the capture in the bank.

   It will allow you to identify whether the payment is an immediate or a deferred payment.

8. 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	Code of the currency used for the payment.
vads_change_rate	Exchange rate used for calculating 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	Currency used for the capture in the bank.

9. Retrieve the value of the **vads_auth_result** field to identify the result of the authorization request.

   The complete list of returned codes can be viewed in the data dictionary.

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

Value	Description
03	**Invalid acceptor** This code is sent by the card issuer. It refers to a configuration problem on authorization servers. (e.g. closed contract, incorrect MCC declared, etc.). **To find out the specific reason of the rejection, the buyer must contact his or her bank.**
05	**Do not honor** This code is sent by the card issuer. This code is used in the following cases: • Invalid expiry date • Invalid CVV • Exceeded credit limit • Insufficient funds (etc.) **To find out the specific reason of the rejection, the buyer must contact his or her bank.**
51	**Insufficient balance or exceeded credit limit** This code is sent by the card issuer. This code appears if the funds on the buyer's account are insufficient for making the purchase. **To find out the specific reason of the rejection, the buyer must contact his or her bank.**
56	**Card absent from the file** This code is sent by the card issuer. The entered card number is incorrect or the card number + expiration date combination does not exist.
57	**Transaction not allowed for this cardholder** This code is sent by the card issuer. This code is used in the following cases: • The buyer attempts to make an online payment with a cash withdrawal card • The authorized payment limit is exceeded **To find out the specific reason of the rejection, the buyer must contact his or her bank.**
59	**Suspected fraud** This code is sent by the card issuer. This code appears when an incorrect CVV code or expiration date has been entered several times. **To find out the specific reason of the rejection, the buyer must contact his or her bank.**
60	**The acceptor of the card must contact the acquirer** This code is sent by the card issuer. It refers to a configuration problem on authorization servers. It is used when the merchant ID does not correspond to the used sales channel (e.g.: an e-commerce transaction with a distant sale contract with manual entry of contract data). **Contact the customer service to resolve the problem.**
81	**Unsecured payment is not accepted by the issuer** This code is sent by the card issuer. After receiving this code, the payment gateway automatically makes a new payment attempt with 3D Secure authentication, when possible.

10. Retrieve the cardholder authentication result. To do this:

a. Retrieve the value of the **vads_threeds_enrolled** field to identify the status of the card enrollment.

Value	Description
Empty	Incomplete 3DS authentication process (3DS disabled in the request, the merchant is not enrolled or the payment method is not eligible for 3DS).
Y	Authentication available, cardholder enrolled.
N	Cardholder not enrolled.
U	Impossible to identify the cardholder or authentication is not available for the card (e.g. commercial or prepaid cards).

b. Retrieve the result of cardholder authentication by retrieving the value of the **vads_threeds_status** field.

Value	Description
Empty	Incomplete 3DS authentication (3DS disabled in the request, the cardholder is not enrolled or the payment method is not eligible for 3DS).
Y	Cardholder authentication success.
N	Cardholder authentication error.
U	Authentication impossible.

Value	Description
A	Authentication attempted but not completed.

11. Retrieve the result of fraud checks by identifying the value of the **vads_risk_control** field. This field is sent only if the merchant has:

- subscribed to the "**Risk management**" service,
- enabled at least one verification process in the Merchant Back Office (**Settings** > **Risk management** menu).

It is populated with the list of values separated by ";" with the following syntax: **vads_risk_control = control1=result1;control2=result2**

The possible values for **control** are:

Value	Description
CARD_FRAUD	Verifies whether the cardholder's card number is on the card greylist.
SUSPECT_COUNTRY	Checks whether the issuing country of the buyer's card is on the list of forbidden countries.
IP_FRAUD	Verifies whether the cardholder's IP address is on the IP greylist.
CREDIT_LIMIT	Checks the purchase frequency and amounts for the same card number, or the maximum amount of an order.
BIN_FRAUD	Checks whether the BIN code of the card is on the BIN code greylist.
ECB	Checks whether the buyer's card is of "e-carte bleue" type.
COMMERCIAL_CARD	Checks whether the buyer's card is a commercial card.
SYSTEMATIC_AUTO	Checks whether the buyer's card is a MAESTRO or VISA ELECTRON card.
INCONSISTENT_COUNTRIES	Checks whether the country of the IP address, the country of the payment card and the buyer's country of residence match.
NON_WARRANTY_PAYMENT	Liability shift.
SUSPECT_IP_COUNTRY	Checks whether the buyer's country, identified by their IP address, is on the list of forbidden countries.

The possible values for **result** are:

Value	Description
OK	OK.
WARNING	Informational control failed.
ERROR	Blocking control failed.

12. Retrieve the card type used for the payment.

Two scenarios are possible:

- For a payment processed with **only one card**. The fields to process are:

Field name	Description
vads_card_brand	Brand of the card used for the payment, e.g.: CB, VISA, VISA_ELECTRON, MASTERCARD, MAESTRO, VPAY
vads_brand_management	Permits to know the brand used when paying, the list of available brands and also if the buyer has changed the default brand chosen by the merchant.
vads_card_number	Card number used for the payment.
vads_expiry_month	Expiry month between 1 and 12 (e.g.: 3 for March, 10 for October).
vads_expiry_year	Expiry year in 4 digits (e.g.: 2023).
vads_bank_code	Code of the issuing bank
vads_bank_label	Name of the issuing bank
vads_bank_product	Product code of the card

Field name	Description
vads_card_country	Country code of the country where the card was issued (alpha ISO 3166-2 code, e.g.: "FR" for France, "PF" for French Polynesia, "NC" for New Caledonia, "US" for the United States).

- For a **split payment** (i.e. a transaction using several payment methods), the following fields must be processed:

Field name	Value	Description
vads_card_brand	MULTI	Several types of payment cards are used for the payment.
vads_payment_seq	In Json format, see details below.	Details of performed transactions.

The **vads_payment_seq** field (json format) describes the split payment sequence. It contains the following elements:

1. "trans_id": transaction identifier used for the entire payment sequence.
2. "transaction": table of sequence transactions. It contains the following elements:

Field name	Description
amount	Amount of the payment sequence.
operation_type	Debit transaction.
auth_number	Authorization number. Will not be returned if not applicable to the used payment method. Example: 949478
auth_result	Return code of the authorization request.
capture_delay	Delay before the capture (in days). • For a payment by card, this parameter is the requested capture date (ISO 8601 format). If not sent in the payment form, the value defined in the Merchant Back Office will be used.
card_brand	Used payment method. For a payment by card (e.g. CB or Visa or MasterCard co-branded CB cards), this parameter is set to **"CB"**. See the Payment Gateway Implementation Guide available in our online documentation archive to see the complete list of card types.
card_number	Payment method number.
expiry_month	Expiry month of the payment method.
expiry_year	Expiry year of the payment method.
payment_certificate	Payment certificate.
contract_used	Contract used for the payment.
identifier	Unique identifier (token) associated with a payment method.
identifier_status	Only present if the requested action is token creation or update. Possible values:

Value	Description
CREATED	The authorization request has been accepted. The token (or UMR for SEPA payment) has been successfully created.
NOT_CREATED	The authorization request has been declined. The token (or UMR for SEPA payment) has not been created, and therefore cannot be viewed in the Merchant Back Office.
UPDATED	The token (or UMR for SEPA payment) has been successfully updated.
NOT_UPDATED	The token (or UMR for SEPA payment) has not been updated.
ABANDONED	The action has been abandoned by the buyer (debtor). The token (or UMR for SEPA payment) has not been created, and therefore cannot be viewed in the Merchant Back Office.

presentation_date	For a payments by card, this parameter is the requested capture date (ISO 8601 format).
trans_id	Transaction number.

<table>
<tr><th>Field name</th><th>Description</th></tr>
<tr><td>ext_trans_id</td><td>This field is not sent for credit card payments.</td></tr>
<tr><td>trans_uuid</td><td>Unique reference generated by the payment gateway after the creation of a payment transaction. Guarantees that each transaction is unique.</td></tr>
<tr><td>extra_result</td><td>Numeric code of the risk assessment result.
<table><tr><th>Code</th><th>Description</th></tr><tr><td>Empty</td><td>No verification completed.</td></tr><tr><td>00</td><td>All the verification processes have been successfully completed.</td></tr><tr><td>02</td><td>Credit card velocity exceeded.</td></tr><tr><td>03</td><td>The card is on the Merchant’s greylist.</td></tr><tr><td>04</td><td>The country of origin of the card is on the Merchant’s greylist.</td></tr><tr><td>05</td><td>The IP address is on the Merchant’s greylist.</td></tr><tr><td>06</td><td>The BIN code is on the Merchant’s greylist.</td></tr><tr><td>07</td><td>Detection of an e-carte bleue.</td></tr><tr><td>08</td><td>Detection of a national commercial card.</td></tr><tr><td>09</td><td>Detection of a foreign commercial card.</td></tr><tr><td>14</td><td>Detection of a card that requires systematic authorization.</td></tr><tr><td>20</td><td>Relevance verification: countries do not match (country IP address, card country, buyer's country).</td></tr><tr><td>30</td><td>The country of the this IP address is on the greylist.</td></tr><tr><td>99</td><td>Technical issue encountered by the server during a local verification process.</td></tr></table></td></tr>
<tr><td>sequence_number</td><td>Sequence number.</td></tr>
<tr><td>trans_status</td><td>Status of the transaction.</td></tr>
</table>

Canceled transactions are also displayed in the table.

13. Store the value of the **vads_trans_uuid** field. It will allow you to assign unique identification to the transaction if you use the Web Service APIs.

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

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

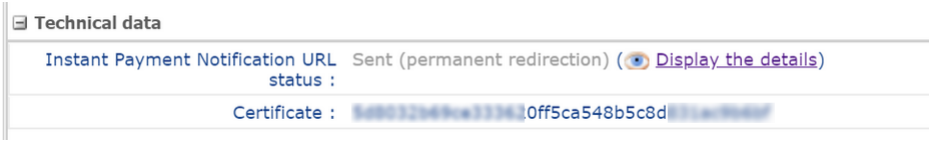
    Their values are identical to the ones submitted in the form.

15. Proceed to order update.

15.7. Running tests and troubleshooting

In order to test the notifications, follow the steps below:

1. Make a payment (in TEST mode or in PRODUCTION mode).
2. Once the payment is complete, look for the transaction in your Back Office (**Management** > **Transactions** or **TEST Transactions** menu if you made the payment in TEST mode).
3. Double-click the transaction to view the **transaction details**.
4. In the transaction details, search for the section entitled **Technical data**.
5. Check the status of the Instant Payment Notification URL:



The list of possible statuses is provided below:

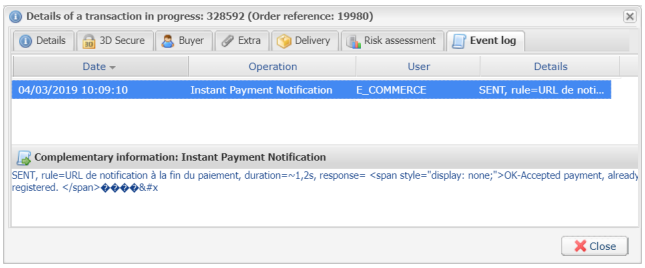
Status	Description
N/A	The transaction did not result in a notification or no notification rules have been enabled.
Undefined URL	An event has triggered the end of payment notification rule but the URL is not configured.
Call in progress	The notification is in progress. This status is temporary.
Sent	The notification has been successfully sent and a remote device returned an HTTP 200, 201, 202, 203, 204, 205 or 206 response status code.
Sent (permanent redirection)	The merchant website has returned an HTTP 301 or 308 response status code with a new URL to contact. A new call in POST mode has been made to the new URL.
Sent (temporary redirection)	The merchant website has returned an HTTP 302 or 307 response status code with a new URL to contact. A new call in POST mode has been made to the new URL.
Sent (redirection to another page)	The merchant website has returned an HTTP 303 response status code with a new URL to contact. A new call in GET mode has been made to the new URL.
Failed	Generic error different from the codes described below.
Server unavailable	The notification has lasted more than 35s.
SSL handshake failure	Your server is incorrectly configured. Run a test on the Qualys website (*https://www.ssllabs.com/ssltest/*) and correct the errors.
Connection interrupted	Communication error.
Connection refused	Communication error.
Server error 300	Case of redirection not supported by the gateway.
Server error 304	Case of redirection not supported by the gateway.
Server error 305	Case of redirection not supported by the gateway.
Server error 400	The merchant website returned a HTTP 400 Bad Request code.
Server error 401	The merchant website returned a HTTP 401 Unauthorized code. Make sure that the resource is not protected by an .htaccess file.
Server error 402	The merchant website returned a HTTP 402 Payment Required code.
Server error 403	The merchant website returned a HTTP 403 Forbidden code. Make sure that the resource is not protected by an .htaccess file.
Server error 404	The merchant website returned a HTTP 404 Not Found code. Make sure that the URL is correctly specified in the rule configuration. Make sure that the file is present on your server.
Server error 405	The merchant website returned a HTTP 405 Method Not allowed code.
Server error 406	The merchant website returned a HTTP 406 Not Acceptable code.
Server error 407	The merchant website returned a HTTP 407 Proxy Authentication Required code.

Status	Description
Server error 408	The merchant website returned a HTTP 408 Request Time-out code.
Server error 409	The merchant website returned a HTTP 409 Conflict code.
Server error 410	The merchant website returned a HTTP 410 Gone code.
Server error 411	The merchant website returned a HTTP 411 Length Required code.
Server error 412	The merchant website returned a HTTP 412 Precondition Failed code.
Server error 413	The merchant website returned a HTTP 413 Request Entity Too Large code.
Server error 414	The merchant website returned a HTTP 414 Request-URI Too long code.
Server error 415	The merchant website returned a HTTP 415 Unsupported Media Type code.
Server error 416	The merchant website returned a HTTP 416 Requested range unsatisfiable code.
Server error 417	The merchant website returned a HTTP 417 Expectation failed code.
Server error 419	The merchant website returned a HTTP 419 Authentication Timeout code.
Server error 421	The merchant website returned a HTTP 421 Misdirected Request code.
Server error 422	The merchant website returned a HTTP 422 Unprocessable Entity code.
Server error 423	The merchant website returned a HTTP 423 Locked code.
Server error 424	The merchant website returned a HTTP 424 Failed Dependency code.
Server error 425	The merchant website returned a HTTP 425 Too Early code.
Server error 426	The merchant website returned a HTTP 426 Upgrade Required code.
Server error 429	The merchant website returned a HTTP 431 Request Header Fields Too Large code.
Server error 431	The merchant website returned a HTTP 415 Unsupported Media Type code.
Server error 451	The merchant website returned a HTTP 451 Unavailable For Legal Reasons code.
Server error 500	The merchant website returned a HTTP 500 Internal Server Error code. An application error has occurred on the level of the server hosting your shop. See the logs of your HTTP server (usually apache). The issue can only be corrected by performing an action on your server.
Server error 501	The merchant website returned a HTTP 501 Not Implemented code.
Server error 502	The merchant website returned a HTTP 502 Bad Gateway / Proxy Error code.
Server error 503	The merchant website returned a HTTP 503 Service Unavailable code.
Server error 504	The merchant website returned a HTTP 504 Gateway Time-out code. The merchant server has not accepted the call within the time limit of 10s.
Server error 505	The merchant website returned a HTTP 505 HTTP Version not supported code.

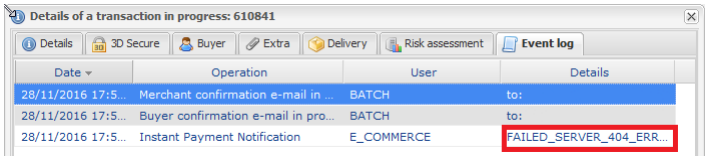
For more information on a notification, click the link **Display the details** or click the **Event log** tab and search for the line **Notification URL call**.

In order to help the merchant identify the source of the error, the gateway systematically analyses the 512 first characters returned by the merchant website and displays them in the **Details** column.

- Example of a successfully processed notification:



- Example of a failed notification:



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

It contains:

- The HTTP code of the encountered error
- Parts of error analysis
- Its consequences
- Instructions to follow via the Merchant Back Office for resending the request to the URL specified in step 4

16. RETURNING TO THE SHOP

By default, when the buyer returns to the merchant website, no parameters will be transmitted by their browser.

However, if the **vads_return_mode** field has been transmitted in the payment form (see chapter **Managing the return to the merchant website**) it will be possible to retrieve the data:

- either via GET, the data is presented in the URL as follows: ?field1=value1&field2=value2
- or via POST: the data is sent in a POST form

The data transmitted to the browser is the same as for notifications (IPN).

The **vads_url_check_src** and **vads_hash** fields will be sent only in the instant notification.

To analyze this data, see chapter **Analyzing the payment result**.

The return to the shop should only allow you to display visual context to the buyer. Do not use the received data for processing in the database.

17. PROCEEDING TO TEST PHASE

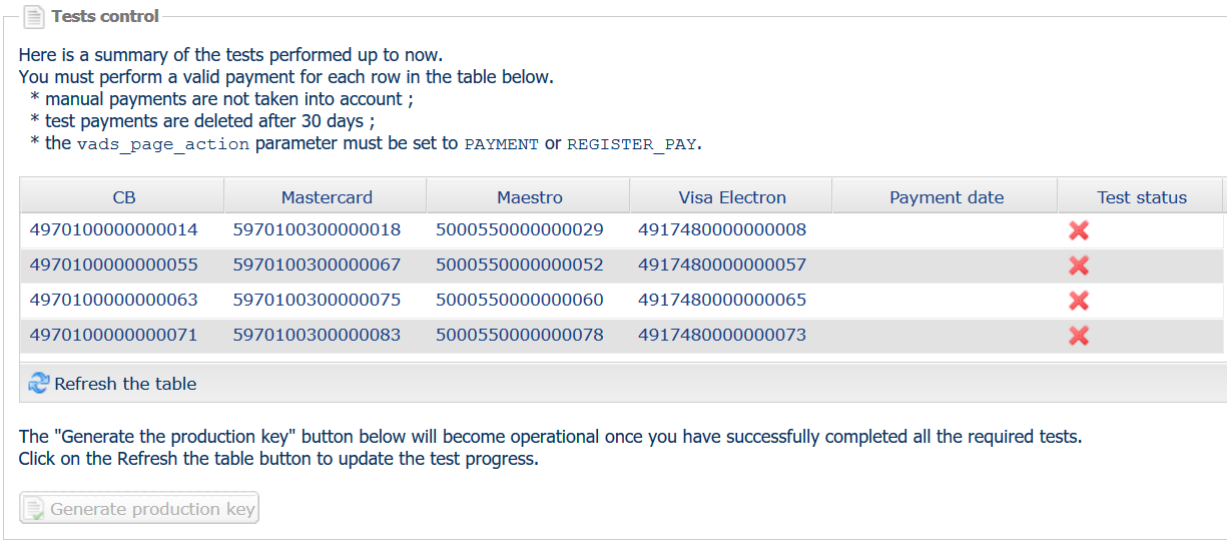
Before the shop goes into production, it is necessary to carry out tests to ensure that the merchant website and the payment gateway are working properly.

The test payment requests must:

- contain the **vads_ctx_mode** field set to **TEST**
- use the **test key** for signature computation

Different cases of payments can be simulated by using test card numbers specified on the payment page. The Merchant will be able to test all 3D Secure authentication results (if the Merchant is enrolled and 3DS is not disabled).

The list of tests to be performed to generate the production key is provided in the Merchant Back Office, **Settings** > **Shop** > **Keys** menu.

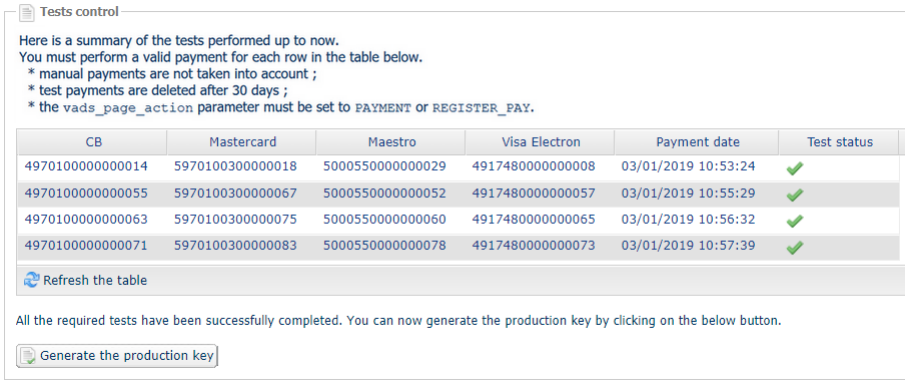


Each row of the list contains card numbers associated with the same scenario (i.e. 2 accepted payments and 2 refused payments).

Each column corresponds to a different card type: CB/VISA, MASTERCARD, MAESTRO, VISA ELECTRON.

To perform the test phase:

1. Make an order on your merchant website as if you were one of your buyers.
2. Once redirected to the payment page, select the card type of your choice.
3. Refer to the list of tests to identify the card number to use.
4. Once a test has been validated, its status is updated on the list. Click the **Refresh the table** button if the status has not been updated automatically.
5. Once the 4 tests have been validated, the **Generate production key** button becomes available.



6. Click the **Generate production key** button and accept the notification messages that will appear.

The production key is now available.

18. ACTIVATING THE SHOP IN PRODUCTION MODE

18.1. Generating the production key

You can generate the production key via **Settings** > **Shop** > **Keys** tab > **Generate the production key** button.

Once the production key has been generated, its value appears in the **Keys** tab.

An e-mail is sent to the company administrator to confirm that the production key has been generated.

18.2. Shifting your merchant website to production mode

1. Set the **vads_ctx_mode** field to **PRODUCTION**.
2. Edit the value of the test key with the value of your production key to compute the signature.
   You will find this value via **Settings** > **Shop** > **Keys** tab.
3. Enter the correct IPN URL in PRODUCTION mode via **Settings** > **Notification rules**.

18.3. Making the first production payment

We recommend checking the two following points:

- The correct end-to-end functioning of the production environment.

  To do that, make a real transaction of at least €2.

  This transaction can later be canceled via the Merchant Back Office, menu: **Management** > **Transactions** > **Transactions is progress** tab. This transaction will not be captured in the bank.

  However, it is recommended to let the transaction be captured at the bank in order to confirm that the merchant's account has been credited. It will then be possible to proceed to a refund.

- The correct functioning of the IPN URL (Instant Payment Notification URL at the end of the payment) specified in the Merchant Back Office.

  To do this, do not click **Return to the shop** after a payment.

  View the transaction details in the Merchant Back Office and make sure that the IPN URL status is **Sent.**

18.4. Regenerating the production key

In case of lost or corrupted production key, the merchant can generate a new one via their Merchant Back Office. To do this:

1. In the Merchant Back Office, select **Settings > Shop > Keys** tab.

2. Click **Regenerate**.

19. OBTAINING HELP

Looking for help? See our FAQ:

https://sogecommerce.societegenerale.eu/doc/fr-FR/faq/faq-homepage.html

For any technical inquiries or if you need any help, contact *technical support*.

In view of facilitating the processing of your requests, please have your shop ID ready (an 8-digit number).

This information is available in the “registration of your shop” e-mail or in the Merchant Back Office (**Settings** > **Shop** > **Configuration**).

20. DATA DICTIONARY

The data dictionary lists all the fields that can be used in a payment form.

First, is presents the main categories (such as technical information, order details, etc.). All the fields that belong to a category are presented.

These tables are presented as follows:

- **Field name**: indicates the name of the parameter as it appears in the HTTP request
- **Format**: data format
- **Description**: description of the field
- **Input**: a field to be transmitted in the request
- **Output**: a field transmitted in the response

The data dictionary also presents the details for each field. Each field is presented as follows:

- **Description**: description of the field
- **Format**: data format (see the table **List of fields and formats** below)
- **Possible values**: expected values when the field must be populated with specific values
- **Example**: example of correct data encoding
- **Error code**: in case there is a error between the merchant website and the payment gateway, the payment gateway indicates the incorrect parameter in the **vads_extra_result** field using a numeric code
- **Note**: additional information, elaboration
- **Category**: category to which the field belongs

Precisions on **error codes:**

An **error code** corresponds to the error number when an incorrect payment form is being submitted.

- In test mode this code will be displayed on the payment page.
- In production mode a warning e-mail will be sent specifying the error code and the name of the incorrect parameter.

Example: Error 09 corresponds to a payment amount error. The submitted amount does not respect the required format.

Viewing parameters sorted by category

Go to the desired category to obtain the list of related parameters

- *3DS Authentication*
- *Recurring payment details*
- *Buyer details*
- *Payment method details*
- *Order details*
- *Shipping details*
- *Technical details*
- *Transaction details*
- *Payment page customization*
- *Automatic redirection*

Technical information

Field name	Format	Description	Input	Output
signature	an40	Signature guaranteeing the integrity of the requests exchanged between the merchant website and the payment gateway.	**x**	**x**
vads_action_mode	enum	Acquisition mode for payment method data	**x**	**x**
vads_override_payment_cinematic	enum	Payment process to apply. Overrides the recorded value on the MID	**x**	
vads_contrib	ans..128	Name of the e-commerce solution used on the merchant website and its version number.	**x**	**x**
vads_ctx_mode	enum	Mode of interaction with the payment gateway	**x**	**x**
vads_extra_result	n2	Risk assessment result (specific to shops that have enabled the feature). Its meaning depends on the value entered in **vads_result**.		**x**
vads_hash	an64	A unique key returned only to the Instant Payment Notification (IPN).		**x**
vads_page_action	enum	Defines the action to be performed.	**x**	**x**
vads_payment_error	n..3	Error codes for a declined payment.		**x**
vads_result	n2	Return code of the requested action.		**x**
vads_site_id	n8	Shop ID	**x**	**x**
vads_url_check	ans..1024	URL of the page to notify at the end of payment. Overrides the value entered in the notification rules settings.	**x**	
vads_url_check_src	enum	This parameter defines the source of the call to the notification URL (also called IPN URL).		**x**
vads_version	enum	Version of the exchange protocol with the payment gateway	**x**	**x**

Order details

Field name	Format	Description	Input	Output
vads_authent_paypal_protection_eligibility	enum	Type of merchant protection used for the transaction.		x
vads_collection_expiry_date	dateTime	Fund collection expiration date used for cash payments.		x
vads_ext_info	ans	Custom fields allowing to add optional fields to the confirmation e-mail sent to the merchant and to the IPN URL.	x	x
vads_ext_info_description	ans..128	Description of the invoice, defined by the merchant during the invoice order creation.		x
vads_ext_info_soft_descriptor	ans	Allows to customize the brand name		x
vads_insurance_amount	n..12	Insurance amount for the entire order.	x	
vads_nb_products	n..12	Number of items in the cart	x	
vads_order_description	ans..65	Order description used for cash payments.	x	x
vads_order_id	ans..64	Order ID	x	x
vads_order_info	ans..255	Additional order info	x	x
vads_order_info2	ans..255	Additional order info	x	x
vads_order_info3	ans..255	Additional order info	x	x
vads_pretax_amount	n..12	Tax-free amount of the entire order		x
vads_product_amountN	n..12	Price of the item incl. VAT. N corresponds to the index of the item (0 for the first one, 1 for the second one, etc.).	x	
vads_product_ext_idN	an..100	Product barcode on the merchant website. N corresponds to the index of the item (0 for the first one, 1 for the second one, etc.).	x	
vads_product_labelN	an..255	Item name. N corresponds to the index of the item (0 for the first one, 1 for the second one, etc.).	x	
vads_product_qtyN	n..12	Item quantity. N corresponds to the index of the item (0 for the first one, 1 for the second one, etc.).	x	
vads_product_refN	an..64	Item reference. N corresponds to the index of the item (0 for the first one, 1 for the second one, etc.).	x	
vads_product_typeN	enum	Item type. N corresponds to the index of the item (0 for the first one, 1 for the second one, etc.).	x	
vads_product_vatN	n..12	Amount or VAT rate applied to the item. N corresponds to the index of the item (0 for the first one, 1 for the second one, etc.).	x	
vads_tax_amount	n..12	Amount of taxes for the entire order	x	
vads_tax_rate	enum	VAT applied to the order	x	x
vads_totalamount_vat	n..12	Total amount of taxes for the entire order	x	x

Buyer details

Field name	Format	Description	Input	Output
vads_avs_result	a1	Address verification system (AVS)		x
vads_cust_address	ans..255	Postal address	x	x
vads_cust_address_number	ans..64	Street number	x	x
vads_cust_address2	ans..255	Address line 2	x	x
vads_cust_cell_phone	an..32	Cell phone number	x	x
vads_cust_city	an..128	City	x	x
vads_cust_country	a2	Country code in compliance with the ISO 3166 alpha-2 standard	x	x
vads_cust_district	ans..127	District	x	x
vads_cust_email	ans..150	Buyer's e-mail address	x	x
vads_cust_first_name	ans..63	First name	x	x
vads_cust_id	an..63	Buyer reference on the merchant website	x	x
vads_cust_last_name	ans..63	Last name	x	x
vads_cust_legal_name	ans..100	Buyer's legal name	x	
vads_cust_name	an..127	Use **vads_cust_first_name** and **vads_cust_last_name**.	x	x
vads_cust_national_id	ans..255	National identifier	x	x
vads_cust_phone	an..32	Phone number	x	x
vads_cust_state	ans..127	State / Region	x	x
vads_cust_status	enum	Status	x	x
vads_cust_title	an..63	Buyer's title	x	x
vads_cust_zip	an..64	Zip code	x	x
vads_ext_info_bil_address_complement	ans..250	Address line 2 specified for billing	x	
vads_ext_info_bil_date_of_birth	Datetime	The buyer's date of birth on the receipt	x	
vads_ext_info_bil_gender	n1	The buyer's gender on the receipt	x	
vads_ext_info_fingerprint_id	string	Unique session identifier	x	
vads_pays_ip	a2	Country code of the buyer's IP address code in the ISO 3166 alpha2 format.		x
vads_proof_of_id_number	an..13	Field reserved to the entry of the buyer's ID number on the payment page.	x	
vads_proof_of_id_type	enum	This field corresponds to the type of ID selected by the buyer.	x	
vads_user_info	ans..255	Information about the user who made the payment.		x

Shipping details

Field name	Format	Description	Input	Output
vads_ext_info_deadline	n	Definition of the delivery delay in days (N days)	**x**	
vads_ext_info_ship_address_complement	ans..250	Address line 2 specified for the shipping	**x**	
vads_ext_info_ship_date_of_birth	Datetime	The buyer's date of birth specified for the shipping	**x**	
vads_ext_info_ship_gender	n1	The buyer's gender specified for the shipping	**x**	
vads_ship_to_city	an..128	City	**x**	**x**
vads_ship_to_country	a2	Country code in compliance with the ISO 3166 standard	**x**	**x**
vads_ship_to_delay	enum	Shipping delay, mandatory for priority shipping	**x**	
vads_ship_to_delivery_company_name	ans..127	Transporter's name	**x**	
vads_ship_to_district	ans..127	District	**x**	**x**
vads_ship_to_first_name	ans..63	First name	**x**	
vads_ship_to_last_name	ans..63	Last name	**x**	
vads_ship_to_legal_name	an..100	Legal name	**x**	
vads_ship_to_name	ans..63	**Deprecated**. Buyer's last name. Use **vads_ship_to_first_name** and **vads_ship_to_last_name**.	**x**	**x**
vads_ship_to_phone_num	ans..32	Phone number	**x**	**x**
vads_ship_to_speed	enum	Shipping speed	**x**	
vads_ship_to_state	ans..127	State / Region	**x**	**x**
vads_ship_to_status	enum	Allows to specify the type of the shipping address.	**x**	**x**
vads_ship_to_street	ans..255	Postal address	**x**	**x**
vads_ship_to_street_number	an..5	Street number	**x**	**x**
vads_ship_to_street2	ans..255	Address line 2	**x**	**x**
vads_ship_to_type	enum	Transport type	**x**	
vads_ship_to_user_info	ans..255	Buyer details (CPF/CNPJ legal identifier)	**x**	**x**
vads_ship_to_zip	an..64	Zip code	**x**	**x**
vads_shipping_amount	n..12	Shipping fee amount	**x**	

Payment method details

Field name	Format	Description	Input	Output
vads_acquirer_network	ans..250	Acquirer network code.		**x**
vads_auth_mode	enum	Mode of the authorization request.		**x**
vads_auth_number	an..20	Authorization number returned by the bank server.		**x**
vads_auth_result	an..11	Return code of the authorization request returned by the issuing bank.		**x**
vads_bank_code	n5	Code associated with the issuing bank.		**x**
vads_bank_label	ans..255	Name of the issuing bank of the payment card.		**x**
vads_bank_product	an..3	Product code of the card used for the payment.		**x**
vads_birth_day	n..2	Date of birth of the cardholder.	**x**	
vads_birth_month	n..2	Month of birth of the cardholder.	**x**	
vads_birth_year	n4	Year of birth of the cardholder.	**x**	
vads_brand_management	json	This field indicates to the merchant: • whether the buyer used a different brand than the default brand defined by the merchant • the brand chosen by the buyer • the list of available brands		**x**
vads_card_brand	an..127	Payment method used, if available (empty otherwise).	**x**	**x**
vads_card_country	a2	Country code alpha-2 (ISO 3166) of the card used for the payment.		**x**
vads_card_holder_name	ans..255	Name of the cardholder.	**x**	
vads_card_nature	a1	Nature of the card.		**x**
vads_card_number	n..36	Masked card number.		**x**
vads_card_product_category	a1	Card product category.		**x**
vads_expiry_month	n..2	Expiration month of the payment card.		**x**
vads_expiry_year	n4	Expiration year of the payment card.		**x**
vads_wallet	an..127	Allows the merchant to identify the type of wallet that was used for the payment.		**x**

Transaction details

Field name	Format	Description	Input	Output
vads_acquirer_payment_reference	ans..250	Payment reference that appears on the payment receipt.		x
vads_acquirer_service_supplier_id	ans..250	Identifier of the entity for which the payment is made.		x
vads_acquirer_transient_data	json	Information specific to the acquirer.	x	
vads_amount	n..12	Transaction amount expressed in the smallest currency unit (cents for euro).	x	x
vads_archival_reference_id	an..25	Reference generated by the payment gateway and transmitted to the acquirer for remittance processing.		x
vads_authent_nsu	ans..255	Unique Sequence Number. Used in Latin America.		x
vads_capture_delay	n..3	Delay in days before capture in the bank.	x	x
vads_change_rate	string	Exchange rate used for calculating the effective payment amount (multi-currency payment).		x
vads_contract_used	ans..250	Merchant ID used for the payment.		x
vads_contracts	map	Merchant ID to be used for the payment.	x	
vads_currency	n3	Code of the currency to use for the payment.	x	x
vads_dcc_amount	n..12	Transaction amount expressed in the smallest unit of the currency chosen by the Buyer if the card is eligible for the Dynamic Currency Conversion (DCC) service.		x
vads_dcc_currency	n3	Numeric code (in accordance with the ISO 4217 standard) of the currency chosen by the Buyer if the card is eligible for the Dynamic Currency Conversion (DCC) service.		x
vads_dcc_markup	Décimal	Sales margin rate applied to the total amount when the card is eligible for the Dynamic Currency Conversion (DCC) service.		x
vads_dcc_rate_date	n14	UTC date and time, in YYYYMMDDHHMMSS format, of obtaining the conversion rate from the Dynamic Currency Conversion service (DCC).		x
vads_effective_amount	n..12	The payment amount presented in the smallest unit of the currency used for the capture in the bank (cents for euro).		x
vads_effective_creation_date	n14	Date of transaction registration in UTC format (GMT+0, 24H) (YYYYMMDDHHMMSS).		x
vads_effective_currency	n3	Currency used for the capture in the bank.		x
vads_ext_trans_id	enum	External transaction reference.		x
vads_first_installment_delay	n..3	Number of deferred months to be used for the first installment of payment in installments.	x	
vads_initial_issuer_transaction_identifier	ans32	Unique transaction reference generated by the issuer. Called a "chaining reference", it is used in MIT transactions to indicate to the issuer that the transaction is part of a series of payments for which the cardholder authenticated himself in the first payment.		x
vads_occurrence_type	enum	Type of sequence. Allows to identify if the transaction is part of a series of payments (subscription or payment in installments).		x

Field name	Format	Description	Input	Output
vads_operation_type	enum	Type of operation: debit, credit (refund), verification.		**x**
vads_payment_cards	liste d'enum	Type of payment method.	**x**	
vads_payment_certificate	an40	Field populated by the payment gateway if the authorization has been **successfully** completed.		**x**
vads_payment_config	enum	Payment type: immediate or installment.	**x**	**x**
vads_payment_option_code	an..5	Code of the used payment option.	**x**	**x**
vads_payment_seq	json	Split payment sequence.		**x**
vads_payment_src	enum	Entry mode for payment method data.	**x**	**x**
vads_presentation_date	n14	Requested capture date.		**x**
vads_requestor	enum	In order to modify the value of the "**Aceite**" field for a Boleto Bancario	**x**	**x**
vads_sequence_number	n..3	Installment payment sequence number.		**x**
vads_tax_refund_amount	n..12	Tax credit amount.		**x**
vads_tid	an..255	Terminal ID. POS identifier defined within the acceptance contract.		**x**
vads_token_id	an..32	Payment order ID associated with the transaction.		**x**
vads_trans_date	n14	Date and time in the GMT+0 (UTC) format on a 24-hour clock (YYYYMMDDHHMMSS).	**x**	**x**
vads_trans_id	an6	Unique ID of a transaction	**x**	**x**
vads_trans_status	enum	Status of the transaction.		**x**
vads_trans_uuid	ans32	Unique transaction reference generated by the payment gateway.		**x**
vads_validation_mode	n1	Transaction validation mode.	**x**	**x**
vads_use_case	ans50	Allows you to specify that this is a payment upon shipment.	**x**	**x**
vads_warranty_result	enum	Liability shift in case of accepted payment.		**x**

Cardholder authentication

Field name	Format	Description	Input	Output
vads_threeds_auth_type	enum	Indicates the authentication type of the cardholder (**CHALLENGE**, **FRICTIONLESS** or **DATA ONLY**) . Returned only if the buyer has correctly authenticated him/herself (vads_threeds_status is “Y” or “A”).		**x**
vads_threeds_cavv	ans..28	Designates the cardholder’s authentication through the ACS. It is populated by the 3DS authentication server (ACS) when the buyer has correctly authenticated him/herself (vads_threeds_status equals "Y" or "A").		**x**
vads_threeds_cavvAlgorithm	an1	Algorithm used by the ACS to generate the CAVV value. It is populated by the 3DS authentication server (ACS) when the buyer has correctly authenticated him/herself (vads_threeds_status equals "Y" or "A").		**x**
vads_threeds_eci	an..2	Indicates the E-Commerce index. It is populated by the 3DS authentication server (ACS) when the buyer has correctly authenticated him/herself (vads_threeds_status equals "Y" or "A").		**x**
vads_threeds_enrolled	a1	Indicates the enrollment status of the cardholder. It is populated by the VISA and MASTERCARD (DS) servers during the 3D Secure authentication process.		**x**
vads_threeds_error_code	n..2	Deprecated. Use **vads_threeds_exit_status**.		**x**
vads_threeds_exit_status	n..2	Final status of 3D Secure authentication. Populated by the payment gateway.		**x**
vads_threeds_mpi	n1	Allows the merchant to indicate their preference when it comes to cardholder authentication. At output, indicates the merchant preference actually transmitted to the issuer.	**x**	
vads_threeds_sign_valid	n1	Indicates the validity of the message containing the authentication result. Populated by the payment gateway.		**x**
vads_threeds_status	a1	Defines the cardholder’s authentication status. Populated by the 3DS authentication server (ACS) during the 3D Secure authentication.		**x**
vads_threeds_xid	ans..28	Indicates the unique 3DS authentication reference. It is populated by the authentication server (ACS) during the 3D Secure authentication process.		**x**

Customizing the payment page

Field name	Format	Description	Input	Output
vads_available_languages	Enum list	Allows to specify the list of languages available on the payment page.	**x**	**x**
vads_language	a2	Defines the language of the payment page (ISO 639-1 standard).	**x**	**x**
vads_iframe_options	json	Allows you to customize the background color and the font of the input fields in iframe mode.	**x**	
vads_shop_name	ans..127	Allows to define the shop name as it appears in the summary payment page, the receipt and the confirmation payment e-mail.	**x**	**x**
vads_shop_url	ans..1024	Allows to override the shop URL that appears on the payment page and in payment confirmation e-mails.	**x**	**x**
vads_theme_config	map	Allows to customize certain elements on the payment page.	**x**	

Redirection to the merchant website

Field name	Format	Description	Input	Output
vads_redirect_error_message	ans..255	Automatic redirection: Message displayed on the payment page prior to redirection after an accepted/ declined payment.	**x**	
vads_redirect_error_timeout	n..3	Automatic redirection: Delay (in seconds) before automatic redirection to the merchant website after an accepted/declined payment.	**x**	
vads_redirect_success_message	ans..255	Automatic redirection: Specifies the message that will appear upon automatic redirection to the merchant website at the end of an accepted payment.	**x**	
vads_redirect_success_timeout	n..3	Automatic redirection: Allows to define a delay in seconds before an automatic redirection to the merchant website at the end of an accepted payment. Its value is between 0 and 300 s.	**x**	
vads_return_mode	enum	Allows to specify the data transmission mode to the URLs of return to the merchant website.	**x**	
vads_url_cancel	ans..1024	URL where the buyer will be redirected after clicking on “Cancel and return to shop” before proceeding to payment.	**x**	
vads_url_check	ans..1024	URL of the page to notify at the end of payment. Overrides the value entered in the notification rules settings.	**x**	
vads_url_check_src	enum	This parameter defines the triggering event of the instant notification (also called IPN).		**x**
vads_url_error	ans..1024	URL where the buyer will be redirected in case of an internal processing error.	**x**	
vads_url_post_wallet	ans..1024	URL using which the merchant will be recalled if a wallet is used during the payment.	**x**	**x**
vads_url_refused	ans..1024	URL where the buyer will be redirected in case of a rejected payment.	**x**	
vads_url_return	ans..1024	Default URL where the buyer will be redirected after clicking on “Return to shop”.	**x**	
vads_url_success	ans..1024	URL where the buyer will be redirected in case of a successful transaction.	**x**	

Recurring payment details

Field name	Format	Description	Input	Output
vads_identifier	ans..50	Unique ID (token or unique mandate reference) associated with a payment method.	**x**	**x**
vads_identifier_previously_registered	bool	Verification of the payment method uniqueness.		**x**
vads_identifier_status	enum	Mandate registration status.		**x**
vads_recurrence_number	n..2	Recurrence number of the recurring payment.		**x**
vads_recurrence_status	enum	Recurrence creation status.		**x**
vads_sub_amount	n..12	Amount of each installment except the ones that will be eventually defined by the **vads_sub_init_amount_number**.	**x**	**x**
vads_sub_currency	n3	Numeric code of the currency to be used for the recurring payment in compliance with the ISO 4217.	**x**	**x**
vads_sub_desc	ans..255	Rule for recurring payments to apply according to the iCalendar RFC5545 specification.	**x**	**x**
vads_sub_effect_date	n8	Subscription start date in the UTC time zone, in YYYYMMDD format.	**x**	**x**
vads_sub_init_amount	n..12	Amount of the first installments of the recurring payment.	**x**	**x**
vads_sub_init_amount_number	n..3	Number of installments for which the amount **vads_sub_init_amount** should be applied.	**x**	**x**
vads_subscription	ans..50	ID of the subscription ID to create.	**x**	**x**

Risk analysis details

Field name	Format	Description	Input	Output
vads_risk_analysis_result	enum	Result of risk assessment analysis performed by an external system (ClearSale, CyberSource, etc.).		x
vads_risk_assessment_result	enum	Result of advanced risk assessment analysis performed by the payment gateway.		x
vads_risk_control	map	Result of risk assessment.		x

Sub-merchant details

Field name	Format	Description	Input	Output
vads_submerchant_adress	ans..255	Address of the sub-merchant. Transmitted by the payment facilitator.	x	x
vads_submerchant_adress2	ans..255	Address line 2 of the sub-merchant. Transmitted by the payment facilitator.	x	x
vads_submerchant_city	an..128	City of the sub-merchant. Transmitted by the payment facilitator.	x	x
vads_submerchant_company_type	ans..60	Company type of the sub-merchant. Transmitted by the payment facilitator.	x	x
vads_submerchant_country	ans..64	Country code of the sub-merchant's address (ISO 3166 alpha-2 standard). Transmitted by the payment facilitator.	x	x
vads_submerchant_facilitatorId	ans..128	Payment Facilitator ID. Transmitted by the payment facilitator.	x	x
vads_submerchant_legal_number	ans..24	Legal Entity Identifier of the sub-merchant. Transmitted by the payment facilitator.	x	x
vads_submerchant_mcc	n4	Merchant Category Code of the sub-merchant. Transmitted by the payment facilitator.	x	x
vads_submerchant_mid	n..64	Merchant ID number of the sub-merchant. Transmitted by the payment facilitator.	x	x
vads_submerchant_name	ans..255	Legal name of the sub-merchant. Transmitted by the payment facilitator.	x	x
vads_submerchant_phone	ans..32	Phone number of the sub-merchant. Transmitted by the payment facilitator.	x	x
vads_submerchant_soft_descriptor	ans..255	Soft descriptor of the sub-merchant that appears on the buyer's bank statement. Transmitted by the payment facilitator.	x	x
vads_submerchant_state	ans..128	Region of the sub-merchant address. Transmitted by the payment facilitator.	x	x
vads_submerchant_url	ans..128	URL of the sub-merchant. Transmitted by the payment facilitator.	x	x
vads_submerchant_zip	ans..64	Zip code of the sub-merchant. Transmitted by the payment facilitator.	x	x

signature

Description Mandatory parameter.

Allows to verify the integrity of transmitted requests.

This value is computed:

- by the merchant website during the payment request,
- by the payment gateway during the response.

Input and output field, returned in the response (IPN and Return URL).

Format an40

Error code **00 - signature** Appears if the value of this field is incorrect,

70-empty params if the field is absent or empty.

Frequent errors:

- The fields of the form have not been encoded in UTF-8.
- The MODE (TEST or PRODUCTION) or the key used is incorrect.
- Line break or carriage return posted in the form.
- Quotation marks ["] posted in the form.
- The type of computation algorithm used is not the correct one.
- The transmitted signature does not respect the rule of signature computation.

Category Technical details

vads_acquirer_network

Description Acquirer network code.

Output field, returned in the response (IPN and Return URL).

Format ans..250

Possible values

Network code	Description
ACCORD_SANDBOX	Oney network (private and gift cards) - sandbox mode
ACCORD	Oney network (private and gift cards)
AMEXGLOBAL	American Express network
AURORE	Cetelem Aurore network (Brand cards and universal Aurore card)
CB	CB network
CONECS	Titre-Restaurant Conecs network
FULLCB	FULL CB Network (Payment in 3 or 4 times without fees by BNPP PF)
MASTERPASS	MasterPass network
PAYCONIQ	Payconiq network
PAYPAL	PayPal network
PAYPAL_SB	PayPal network - sandbox mode
SEPA	SEPA network (SDD and SCT)

Category Payment method details.

vads_acquirer_payment_reference

Description	Payment reference that appears on the payment receipt. Used in particular for the Multibanco payment method. *Output field, returned in the response (IPN only).*
Format	ans..250
Category	Transaction details.

vads_acquirer_service_supplier_id

Description	Identifier of the entity for which the payment is made. Used in particular for the Multibanco payment method. *Output field, returned in the response (IPN only).*
Format	ans..250
Category	Transaction details.

vads_acquirer_transient_data

Description

Allows to transmit specific information to one or more networks.

An error will be sent upon submission of the form if the specified value does not respect the rules established by the acquirer.

For a Conecs transaction, this field allows to transmit the amount eligible for Meal Voucher (Titre-Restaurant) payment.

Input field.

Format

json

Possible values

For a transaction via Conecs meal voucher, the expected JSON format is:

```
vads_acquirer_transient_data={"CONECS":{"eligibleAmount":"1725"}}
```

For a FULLCB transaction, the expected JSON format is:

```
vads_acquirer_transient_data={"FULLCB":{"data":"DE=3#VT=12#MT=A"}}
```

To limit the cards accepted for payment based on the BIN code, the expected format is as follows:

```
{"MULTI":{"bins":["bin1","bin2","bin3"]}}
```

NB: Supports 6-digit BIN codes or 8-digit BIN codes.

Example:

6-digit BIN code: 4012 34XX XXXX XXXX;

8-digit BIN code: 4000 1234 XXXX XXXX.

Error codes

130, 133, 134, 135, 136, 137, 175, 176

Category	Transaction details.

vads_action_mode

Description	Mandatory parameter. Acquisition mode for payment method data. *Input and output field, returned in the response (IPN and Return URL).*
Format	enum
Error code	47
Possible values	**INTERACTIVE**: the buyer enters the payment method details on the payment page. **IFRAME**: card detail entry on a simplified and streamlined payment page that the merchant can embed into the web page of their choice.
Category	Technical information.

vads_amount

Description	Transaction amount expressed in the smallest currency unit (cents for euro). Example: for a transaction of 10 euros and 28 cents, the value of the parameter is 1028. The payment form will be rejected in the following cases: • an amount equal to zero [vads_amount=0], • a negative amount [vads_amount=-100], • an amount with decimals or points |vads_amount=100.50], • a form without the vads_amount field (amount absent). A message notifying of a technical error will be associated with a **09** return code (**vads_extra_result**). *Input and output field, returned in the response (IPN and Return URL).*
Format	n..12
Error code	09
Category	Transaction details.

vads_archival_reference_id

Description	A reference generated by the payment gateway and sent to the acquirer for remittance processing and transaction reconciliation. Provided only for CB, AMEX andPAYPAL payments. E.g.: L18500026501 *Output field, returned in the response (IPN only).*
Format	an..25
Category	Transaction details.

vads_auth_mode

Description	Specifies the mode of the authorization request. *Output field, returned in the response (IPN and Return URL).*
Format	enum
Possible values	**FULL**: corresponds to an authorization for the total transaction amount. Value used for immediate payments if the period between the requested capture date and the current date is strictly shorter than the authorization validity period. **MARK**: corresponds to an authorization for EUR 1 (or information request about the CB network if the acquirer supports it). Value used for deferred payments if the period between the requested capture date and the current date is strictly greater than the authorization validity period.
Category	Payment method details.

vads_auth_number

Description	Authorization number returned by the authorization server, if available (otherwise, empty). *Output field, returned in the response (IPN and Return URL).*
Format	an..20
Category	Payment method details.

vads_auth_result

Description Return code of the authorization request returned by the issuing bank, if available.

Output field, returned in the response (IPN and Return URL).

Format an..3

Possible values

Codes returned by the **CB** network:

Value	Description	Value	Description
00	Approved or successfully processed transaction	**54**	Expired card
02	Contact the card issuer	**55**	Incorrect secret code
03	Invalid acceptor	**56**	Card absent from the file
04	Keep the card	**57**	Transaction not allowed for this cardholder
05	Do not honor	**58**	Transaction not allowed for this cardholder
07	Keep the card, special conditions	**59**	Suspected fraud
08	Confirm after identification	**60**	The acceptor of the card must contact the acquirer
12	Incorrect Transaction Code	**61**	Withdrawal limit exceeded
13	Invalid amount	**63**	Security rules unfulfilled
14	Invalid cardholder number	**65**	Exceeded number of withdrawals
15	Unknown issuer	**68**	Response not received or received too late
17	Canceled by the buyer	**75**	Number of attempts for entering the secret code has been exceeded
19	Retry later	**76**	The cardholder is already blocked, the previous record has been saved
20	Incorrect response (error on the domain server)	**78**	Transaction blocked, first transaction on card not properly unblocked
24	Unsupported file update	**80**	Contactless payment is not accepted by the issuer
25	Unable to locate the registered elements in the file	**81**	Unsecured payment is not accepted by the issuer
26	Duplicate registration, the previous record has been replaced	**82**	CVV, dCVV, iCVV incorrect
27	File update edit error	**83**	Revocation of all recurring payments for the card
28	Denied access to file	**84**	R1 - Revocation of recurring payment for the card of a specific Merchant or for the MCC and the card
29	Unable to update	**86**	6P - Failure of the issuer to verify the data
30	Format error	**88**	A4 - Misuse of the TRA exemption
31	Unknown acquirer company ID	**90**	Temporary shutdown
33	Expired card	**91**	Unable to reach the card issuer
34	Suspected fraud	**94**	Duplicate transaction
38	Expired card	**96**	System malfunction
41	Lost card	**97**	Overall monitoring timeout
43	Stolen card	**98**	Server not available, new network route requested
46	Customer account closed	**99**	Initiator domain incident
51	Insufficient balance or exceeded credit limit		

Codes returned by **Amex Global** acquirer:

Code	Description
000	Approved
001	Approved with an ID
002	Partial approval (Prepaid Cards only)
100	Declined
101	Expired card / Invalid expiry date
106	Exceeded PIN entry attempts
107	Please Call Issuer
109	Invalid merchant
110	Incorrect Transaction Amount
111	Invalid account / Invalid MICR (Travelers Cheque)
115	Requested function not supported
117	Invalid PIN
119	Cardholder not enrolled / not allowed
122	Invalid card security code (a.k.a., CID, 4DBC, 4CSC)
125	Invalid effective date
130	Declined
181	Format error
183	Invalid currency code
187	Deny - New card issued
189	Deny - Account canceled
200	Deny - Pick up card
900	Accepted - ATC Synchronization
909	System malfunction (cryptographic error)
912	Issuer not available

Codes returned by the **ONEY_API** network:

Code	Description
0	Awaiting acceptance by Oney
1	Payment rejected by Oney
2	Payment accepted by Oney
3	Payment abandoned
4	Payment canceled
99	Unknown error

Codes returned by the **PayPal** network:

Code	Description
0	Transaction accepted
10001	Internal error
10002	Restricted Account
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 in the same currency as the original transaction. • This transaction has already been fully refunded. • The time limit (60 days) for performing a refund for this transaction has been exceeded.
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*.

Other return codes	For payment methods that are different from the ones presented below: • see the technical documentation specific to the payment method or • contact the technical support for more information.
Category	Payment method details.

vads_authent_paypal_protection_eligibility

Description

Type of merchant 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.

Concerns only the PayPal payment method.

Output field, returned in the response (IPN and Return URL).

Format enum

Category Order details.

vads_authent_nsu

Description

Unique sequence number (Latin America).

Output field, returned in the response (IPN and Return URL).

Format ans..255

Category Transaction details.

vads_available_languages

Description

Allows to specify the list of languages available on the payment page.

The elements on the list must be separated by a semi-colon (**;**).

Allows you to modify the list of languages offered by the language selector at the top right of the payment page.

Input and output field, returned in the response (IPN and Return URL).

Format language1;language2;language3

Error code 71

Possible values

Language	Value	Default available language
German	de	x
English	en	x
Chinese	zh	x
Spanish	es	x
French	fr	x
Italian	it	x
Japanese	ja	x
Dutch	nl	x
Polish	pl	
Portuguese	pt	x
Russian	ru	x
Swedish	sv	x
Turkish	tr	

E.g.: to limit the choice to French and English, submit vads_available_languages=fr;en

Category Payment page customization

vads_avs_result

Description

Transmits the result of the address verification performed by the buyer.

This verification only applies to the numeric part of the billing address.

The Address Verification Service is supported in the USA, Canada and United Kingdom.

Output field, returned in the response (IPN and Return URL).

Format a1

Possible values

Code	Visa	MasterCard	Discover	American Express
Y	Address & 5-digit or 9-digit ZIP match	Address & 5-digit ZIP match	Address only matches	Address & ZIP match
A	Address matches, ZIP does not	Address matches, ZIP does not	Address & 5-digit ZIP match	Address only matches
S	AVS not supported	AVS not supported	AVS not supported	AVS not supported
R	System unavailable, retry	System unavailable, retry	Not applicable	System unavailable, retry

Code	Visa	MasterCard	Discover	American Express
U	Information not available	Information not available	System unavailable, retry	Information not available
Z	Either 5-digit or 9-digit ZIP match, address does not	5-digit ZIP matches, address does not	5-digit ZIP matches, address does not	ZIP code only matches
N	Neither ZIP nor address match	Neither ZIP nor address match	Neither ZIP nor address match	Neither ZIP nor address match
W	Not applicable	For U.S., 9-digit ZIP matches, address does not. For non-U.S., ZIP matches, address does not	Information not available	Not applicable
X	Not applicable	For U.S., all digits match. For non-U.S., ZIP and address match.	Address & 9-digit ZIP match	Not applicable
B	Address matches, ZIP not verified	Not applicable	Not applicable	Not applicable
T	Not applicable	Not applicable	9-digit ZIP matches, address does not	Not applicable
P	ZIP matches, address not verified	Not applicable	Not applicable	Not applicable
C	Address and ZIP not verified	Not applicable	Not applicable	Not applicable
D	Address & ZIP match (International only)	Not applicable	Not applicable	Not applicable
G	Address not verified (International only)	Not applicable	Not applicable	Not applicable
I	Address not verified (International only)	Not applicable	Not applicable	Not applicable
M	Address & ZIP match (International only)	Not applicable	Not applicable	Not applicable
F	Address & ZIP match (UK only)	Not applicable	Not applicable	Not applicable

Category Transaction details.

vads_bank_code

Description Code associated with the issuing bank.

Output field, returned in the response (IPN and Return URL).

Format n5

Category Payment method details.

vads_bank_label

Description Name of the issuing bank of the payment card.

Output field, returned in the response (IPN and Return URL).

Format ans..255

Alphanumeric and special characters that may contain accented characters (except '<' and '>').

Category Payment method details

vads_bank_product

Description Product code of the card used for the payment.

Output field, returned in the response (IPN and Return URL).

Format an..20

Possible values

VISA	Designation
A	Visa Traditional
B	Visa Traditional Rewards
C	Visa Signature
D	Visa Signature Preferred
E	Proprietary ATM
F	Visa Classic
G	Visa Business
G1	Visa Signature Business
G2	Reserved
G3	Visa Business Enhanced
G4	Visa Infinite Business
G5	Visa Business Rewards
H	Reserved
I	Visa Infinite
I1	Visa Infinite Privilege
I2	Visa Ultra High Net Worth
J	Reserved
J1	Reserved
J2	Reserved
J3	Visa Healthcare
J4	Reserved
K	Visa Corporate T&E

VISA	Designation
K1	Visa GSA Corporate T&E
L	Electron
N	Visa Platinium
N1	TBA
N2	Visa Select
P	Visa Gold
Q	Private Label
Q1	Reserved
Q2	Private Label Basic
Q3	Private Label Standard
Q4	Private Label Enhanced
Q5	Private Label Specialized
Q6	Private Label Premium
R	Proprietary
S	Visa Purchasing
S1	Visa Purchasing
S2	Visa Purchasing
S3	Visa Purchasing
S4	Government Services Loan
S5	Commercial Transport EBT
S6	Business Loan
S7	Visa Distribution
T	Reserved
U	Visa TravelMoney
V	Visa VPay
W	Reserved
X	Reserved
Y	Reserved
Z	Reserved

MASTERCARD	Designation
BPD	MASTERCARD BUSINESS PREMIUM DEBIT
CIR	CIRRUS
DAG	GOLD DEBIT MASTERCARD SALARY
DAP	PLATINUM DEBIT MASTERCARD SALARY
DAS	STANDARD DEBIT MASTERCARD SALARY
DDB	DOMESTIC DEBIT BRAND
DLG	DEBIT GOLD DELAYED DEBIT
DLH	DEBIT WORLD EMBOSSED DELAYED DEBIT
DLP	DEBIT PLATINUM DELAYED DEBIT
DLS	MASTERCARD CARD-DELAYED DEBIT
DOS	STANDARD DEBIT MASTERCARD SOCIAL
DWF	DEBIT MASTERCARD HUMANITARIAN PREPAID
M	MASTERCARD
MAB	WORLD ELITE MASTERCARD
MAC	MASTERCARD CORPORATE WORLD ELITE
MAP	MASTERCARD COMMERCIAL PAYMENTS ACCOUNT
MBB	MASTERCARD PREPAID CONSUMER
MBC	MASTERCARD PREPAID VOUCHER
MBD	MASTERCARD PROFESSIONAL DEBIT BUSINESS CARD
MBE	MASTERCARD ELECTRONIC BUSINESS CARD
MBK	MASTERCARD BLACK
MBP	MASTERCARD UNKNOWN PRODUCT
MBS	MASTERCARD B2B PRODUCT
MBT	MASTERCARD CORPORATE PREPAID TRAVEL
MBW	WORLD MASTERCARD BLACK EDITION – DEBIT
MCB	MASTERCARD BUSINESS CARD
MCC	MASTERCARD CREDIT MIXED BIN CARD
MCD	MASTERCARD DEBIT CARD
MCE	MASTERCARD ELECTRONIC CARD
MCF	MASTERCARD FLEET CARD
MCG	MASTERCARD GOLD CARD
MCH	MASTERCARD PREMIUM CHARGE
MCO	MASTERCARD CORPORATE CARD
MCP	MASTERCARD PURCHASING CARD
MCS	MASTERCARD STANDARD CARD
MCT	TITANIUM MASTERCARD CARD
MCV	MERCHANT BRANDED PROGRAM
MCW	WORLD MASTERCARD CARD
MDB	DEBIT MASTERCARD BUSINESSCARD CARD
MDG	DEBIT GOLD MASTERCARD CARD
MDH	DEBIT OTHER EMBOSSED
MDJ	DEBIT OTHER 2 EMBOSSED
MDL	BUSINESS DEBIT OTHER EMBOSSED
MDN	BUSINESS DEBIT OTHER 2 EMBOSSED
MDO	DEBIT OTHER CARD
MDP	DEBIT PLATINUM CARD
MDR	DEBIT BROKERAGE CARD
MDS	DEBIT MASTERCARD CARD
MDT	MASTERCARD BUSINESS DEBIT
MDW	WORLD ELITE DEBIT MASTERCARD
MEB	MASTERCARD EXECUTIVE BUSINESS CARD
MEC	MASTERCARD ELECTRONIC COMMERCIAL CARD

MASTERCARD	Designation
MEF	ELECTRONIC PAYMENT ACCOUNT
MEO	MASTERCARD CORPORATE EXECUTIVE CARD
MET	TITANIUM DEBIT MASTERCARD CARD
MFB	FLEX WORLD ELITE
MFD	FLEX PLATINUM
MFE	FLEX CHARGE WORLD ELITE
MFH	FLEX WORLD
MFL	FLEX CHARGE PLATINUM
MFW	FLEX CHARGE WORLD
MGF	MASTERCARD GOUVERNMENT COMMERCIAL CARD
MHA	MASTERCARD HEALTHCARE PREPAID NON-TAX
MHB	MASTERCARD HSA SUBSTANTIATED (DEBIT MASTERCARD)
MHD	HELOC DEBIT STANDARD
MHH	MASTERCARD HSA NON-SUBSTANTIATED (DEBIT MASTERCARD)
MHL	HELOC DEBIT GOLD
MHM	HELOC DEBIT PLATINUM
MHN	HELOC DEBIT PREMIUM
MIA	PREPAID MASTERCARD UNEMBOSSED STUDENT CARD
MIP	PREPAID DEBIT MASTERCARD STUDENT CARD
MIU	DEBIT MASTERCARD UNEMBOSSED
MLA	MASTERCARD CENTRAL TRAVEL SOLUTIONS AIR CARD
MLD	MASTERCARD DISTRIBUTION CARD
MLL	MASTERCARD CENTRAL TRAVEL SOLUTIONS LAND CARD
MNF	MASTERCARD PUBLIC SECTOR COMMERCIAL CARD
MNW	MASTERCARD NEW WORLD
MOC	MASTERCARD UNKNOWN PRODUCT
MOG	MAESTRO GOLD
MOP	MAESTRO PLATINIUM
MOW	MAESTRO WORLD
MPA	MASTERCARD PREPAID DEBIT STANDARD-PAYROLL
MPB	PREFERRED BUSINESS CARD
MPC	MPC
MPD	MASTERCARD FLEX PREPAID
MPF	MASTERCARD PREPAID DEBIT STANDARD-GIFT
MPG	MASTERCARD UNEMBOSSED PREPAID STUDENT CARD
MPH	MASTERCARD CASH PREPAID
MPJ	PREPAID DEBIT MASTERCARD CARD GOLD
MPK	PREPAID MASTERCARD GOUVERNMENT COMMERCIAL CARD
MPL	PLATINIUM MASTERCARD CARD
MPM	MASTERCARD PREPAID DEBIT STANDARD-CONSUMER INCENTIVE
MPN	MASTERCARD PREPAID DEBIT STANDARD-INSURANCE
MPO	MASTERCARD PREPAID DEBIT STANDARD-OTHER
MPP	PRE-PAID CARD
MPR	MASTERCARD PREPAID DEBIT STANDARD-TRAVEL
MPT	MASTERCARD PREPAID DEBIT STANDARD-TEEN
MPV	MASTERCARD PREPAID DEBIT STANDARD-GOVERNMENT
MPW	DEBIT MASTERCARD BUSINESS CARD PREPAID WORK B2B
MPX	MASTERCARD PREPAID DEBIT STANDARD-FLEX BENEFIT
MPY	MASTERCARD PREPAID DEBIT STANDARD-EMPLOYEE INCENTIVE
MPZ	MASTERCARD PREPAID DEBIT STANDARD – GOVERNMENT CONSUMER
MRC	MASTERCARD ELECTRONIC CONSUMER PREPAID
MRF	MASTERCARD EUROPEAN REGULATED INDIVIDUAL PAY

MASTERCARD	Designation
MRG	MASTERCARD STANDARD PREPAID
MRH	MASTERCARD UNKNOWN PRODUCT
MRJ	PREPAID MASTERCARD GOLD CARD
MRK	PREPAID MASTERCARD PUBLIC SECTOR COMMERCIAL CARD
MRL	PREPAID MASTERCARD ELECTRONIC COMMERCIAL CARD (NON-US)
MRO	MASTERCARD REWARDS ONLY
MRP	STANDARD RETAILER CENTRIC PAYMENTS
MRW	MASTERCARD CREDIT BUSINESS CARD PREPAID
MSA	PREPAID MAESTRO PAYROLL CARD
MSB	MAESTRO SMALL BUSINESS CARD
MSF	PREPAID MAESTRO GIFT CARD
MSG	PREPAID MAESTRO CONSUMER RELOADABLE CARD
MSI	MAESTRO
MSJ	PREPAID MAESTRO GOLD
MSM	PREPAID MAESTRO CONSUMER PROMOTION CARD
MSN	PREPAID MAESTRO INSURANCE CARD
MSO	PREPAID MAESTRO OTHER CARD
MSQ	RESERVED FOR FUTURE USE
MSR	PREPAID MAESTRO TRAVEL CARD
MST	PREPAID MAESTRO TEEN CARD
MSV	PREPAID MAESTRO GOVERNMENT BENEFIT CARD
MSW	PREPAID MAESTRO CORPORATE CARD
MSX	PREPAID MAESTRO FLEX BENEFIT CARD
MSY	PREPAID MAESTRO EMPLOYEE INSENTIVE CARD
MSZ	PREPAID MAESTRO EMERGENCY ASSISTANCE CARD
MTP	MASTERCARD PLATINUM PREPAID TRAVEL (UK AND BRAZIL)
MUW	WORLD DOMESTIC AFFLUENT
MWB	WORLD MASTERCARD FOR BUSINESS
MWD	WORLD DEFERRED
MWE	MASTERCARD WORLD ELITE
MWF	MASTERCARD HUMANITARIAN PREPAID
MWO	MASTERCARD CORPORATE WORLD
MWR	WORLD RETAILER CENTRIC PAYMENTS
OLB	MAESTRO SMALL BUSINESS DELAYED DEBIT
OLG	MAESTRO GOLD DELAYED DEBIT
OLP	MAESTRO PLATINUM DELAYED DEBIT
OLS	MAESTRO-DELAYED DEBIT
OLW	MAESTRO WORLD DELAYED DEBIT
PVA	PRIVATE LABEL A
PVB	PRIVATE LABEL B
PVC	PRIVATE LABEL C
PVD	PRIVATE LABEL D
PVE	PRIVATE LABEL E
PVF	PRIVATE LABEL F
PVG	PRIVATE LABEL G
PVH	PRIVATE LABEL H
PVI	PRIVATE LABEL I
PVJ	PRIVATE LABEL J
PVL	PRIVATE LABEL CARD
SAG	GOLD MASTERCARD SALARY–IMMEDIATE DEBIT
SAL	STANDARD MAESTRO SALARY
SAP	PLATINUM MASTERCARD SALARY–IMMEDIATE DEBIT

MASTERCARD	Designation
SAP	PLATINUM MASTERCARD SALARY IMMEDIATE DEBIT
SAS	STANDARD MASTERCARD SALARY–IMMEDIATE DEBIT
SOS	STANDARD MASTERCARD SOCIAL–IMMEDIATE DEBIT
SUR	PREPAID MASTERCARD UNEMBOSSED (NON-US)
SUR	PREPAID UNEMBOSSED MASTERCARD CARD (NON-US)
TBE	MASTERCARD ELECTRONIC BUSINESS IMMEDIATE DEBIT
TCB	MASTERCARD BUSINESS CARD-IMMEDIATE DEBIT
TCC	MASTERCARD MIXED BIN-IMMEDIATE DEBIT
TCE	MASTERCARD ELECTRONIC IMMEDIATE DEBIT
TCF	MASTERCARD FLEET CARD IMMEDIATE DEBIT
TCG	LD MASTERCARD CARD-IMMEDIATE DEBIT
TCO	MASTERCARD (CORPORATE) IMMEDIATE DEBIT
TCP	MASTERCARD PURCHASING CARD IMMEDIATE DEBIT
TCS	MASTERCARD STANDARD CARD-IMMEDIATE DEBIT
TCW	WORLD SIGNIA MASTERCARD CARD-IMMEDIATE DEBIT
TEB	MASTERCARD EXECUTIVE BUSINESS CARD IMMEDIATE DEBIT
TEC	MASTERCARD ELECTRONIC COMMERCIAL IMMEDIATE DEBIT
TEO	MASTERCARD CORPORATE EXECUTIVE IMMEDIATE DEBITCARD
TIU	TIU
TNF	MASTERCARD PUBLIC SECTOR COMMERCIAL CARD IMMEDIATE DE
TNW	MASTERCARD NEW WORLD-IMMEDIATE DEBIT
TPB	PREFERRED BUSINESS CARD IMMEDIATE DEBIT
TPL	PLATINUM MASTERCARD IMMEDIATE DEBIT
TWB	WORLD MASTERCARD BLACK EDITION IMMEDIATE DEBIT
WBE	MASTERCARD UNKNOWN PRODUCT
WDR	WORLD DEBIT MASTERCARD REWARDS
WMR	WORLD MASTERCARD REWARDS

CB	Designation
1	National cash withdrawal card
2	National cash withdrawal and payment card
3	National payment card
4	National payment and cash withdrawal card requiring systematic authorization
5	National payment card requiring systematic authorization

Other product codes	Designation
AX	AMERICAN EXPRESS
DI	DISCOVER
DN	DINERS
JC	JCB

Category Payment method details.

vads_birth_day

Description Date of birth of the cardholder.

Input field.

Format n..2

Error code 76

Category	Payment method details.

vads_birth_month

Description	Month of birth of the cardholder. *Input field.*
Format	n..2
Error code	76
Category	Payment method details.

vads_birth_year

Description	Year of birth of the cardholder. *Input field.*
Format	n4
Error code	76
Category	Payment method details.

vads_brand_management

Description	Indicates to the merchant: • whether the user has chosen a brand (**userChoice** attribute), • the brand chosen by the buyer (**brand** attribute), • the list of available brands (**brandList** attribute). This field is returned only if brand selection is enabled for the CB contract used for the payment. *Output field, returned in the response (IPN and Return URL).*
Format	json
Possible values	Example: vads_brand_management={"userChoice":true,"brand":"CB","brandList":"CB|VISA"}
Category	Payment method details.

vads_capture_delay

Description	Indicates the delay (in days) before the capture. If the parameter is not submitted, the default value specified in the Merchant Back Office will be used. The default value can be configured in the Merchant Back Office by all authorized persons.

- *The value of **vads_capture_delay** is not taken into account in the case of payment in installments **MULTI_EXT**.*

- *If the capture delay is higher than 365 days in the payment request, it will be automatically reset to 365 days.*

Input and output field (IPN and Return URL).

Format	n..3
Error code	06
Category	Transaction details.

vads_card_brand

Description	Payment method used, if available (empty otherwise). The value is derived from the BIN range files. *Output field, returned in the response (IPN and Return URL).*
Format	an..127
Possible values	See the **vads_payment_cards** field. The value CB will be returned for co-branded Visa and MasterCard CB cards.
Category	Payment method details.

vads_card_country

Description	Country code of the card in compliance with the ISO 3166 standard *Output field, returned in the response (IPN and Return URL).*
Format	ISO 3166
Category	Payment method details.

vads_card_holder_name

Description	Name of the cardholder. In Latin America, this parameter is required for DECIDIR and VISANET. *Input field.*
Format	ans..255
Error code	45
Category	Payment method details.

vads_card_nature

Description	Nature of the card. Empty field if not provided by acquirer. *Output field, returned in the response (IPN and Return URL).*
Format	enum

Possible values

- `CONSUMER_CARD`: Personal card
- `COMMERCIAL_CARD`: Commercial card

Category Payment method details.

vads_card_number

Description

- Masked card number. Contains the 6 first digits of the number followed by "XXXXXX" and the 4 last numbers in the end.
- IBAN and BIC used for the payment separated by "_" in case of a direct debit payment.

  The BIC is optional so the number may be the IBAN only.

Output field returned in the response (IPN and Return URL).

Format an..36

Category Payment method details.

vads_card_product_category

Description

Card product category.

The value is derived from the BIN range files.

Output field, returned in the response (IPN and Return URL).

Format enum

Possible values

- `CREDIT`
- `DEBIT`
- `PREPAID`

Category Payment method details.

vads_change_rate

Description

Exchange rate used to calculate the effective payment amount (multi-currency payment).

Output field, returned in the response (IPN and Return URL).

Format string

Category Transaction details

vads_collection_expiry_date

Description

Fund collection expiration date in dateTime format.

Used in Latin America for cash payments.

Allows to override the expiration date of fund collection defined by default in the MID configuration.

E.g.: 2020-02-25T14:02:17+00:00

Input and output field, returned in the response (IPN and Return URL).

Format	dateTime
Error code	174
Category	Order details.

vads_contracts

Description

Allows to:

- specify a list with the Merchant ID (MID) to use for each acceptance network,
- exclude a network.

This parameter is optional and is only used when **you have several e-commerce Merchant IDs (MID)** within the same network and when you wish to **select a different Merchant ID (MID) depending on the payment**.

If this parameter is not specified or absent, the payment will be made with the Merchant ID(s) according to the priority order defined in the Merchant Back Office (**Settings** > **Shop** > **MID association** tab).

Input field.

Format

map

Error code

62

Possible values

Syntax:

- To **define a list** of MIDs, separate them with a semi-colon ";".

```
vads_contracts=NETWORK_CODE_A=MID_A1;NETWORK_CODE_B=MID_B2
```

- To **exclude a network**, add **network name**=**NO**.

```
vads_contracts=NETWORK_CODE_A=NO
```

- To **force the TID**, separate the MID number and the TID number by a colon: ": "

```
vads_contracts=NETWORK_CODE_A=MID_A1:TID_1
```

- If the field is submitted empty, the MID used will be the one defined by the priority order in the Merchant Back Office (**Settings** > **Shop** > **MID association** tab).

The possible networks are:

Network code	Description
ACCORD_SANDBOX	Oney network (private and gift cards) - sandbox mode
ACCORD	Oney network (private and gift cards)
AMEXGLOBAL	American Express network
AURORE	Cetelem Aurore network (Brand cards and universal Aurore card)
CB	CB network
CONECS	Titre-Restaurant Conecs network
FULLCB	FULL CB Network (Payment in 3 or 4 times without fees by BNPP PF)
MASTERPASS	MasterPass network
PAYCONIQ	Payconiq network
PAYPAL	PayPal network
PAYPAL_SB	PayPal network - sandbox mode
SEPA	SEPA network (SDD and SCT)

Examples:

In order to force the MID to be used:

```
vads_contracts="CB=1231231;AMEXGLOBAL=949400444"
```

In order to forbid the payment on a specific network:

```
vads_contracts="CB=1231231;AMEXGLOBAL=NO"
```

Category Transaction details.

vads_contract_used

Description This field defines the value of the Merchant ID (MID) associated with the transaction. It is populated with the Merchant ID (MID) registered by default in your shop or it takes the value of the **vads_contracts** field sent in the payment request.

Output field, returned in the response (IPN and Return URL).

Format ans..250

Category Transaction details

vads_contrib

Description Optional information that indicates the name of the CMS used for the payment (Joomla, osCommerce, etc.). If you are developing your own software, this field can include your own module version for example.

Input and output field, returned in the response (IPN and Return URL).

Format ans..128

Error code 31

Category Technical details

vads_ctx_mode

Description	Mandatory parameter. Defines the mode of interaction with the payment gateway. Affects the choice of the key to be used (test or production) during signature computation. The TEST mode is available at all times, even after the generation of the production key. If you create a new e-commerce website (or have access to the acceptance testing environment), you can make tests without impacting the website that is currently in production. *The input and output field, returned in the response (IPN and Return URL).*
Format	enum
Error code	11 **Frequent errors**: • The mode has not been submitted to the payment gateway. • Using PROD instead of PRODUCTION. • Writing the value in lowercase letters (test or production). This field expects values only in uppercase letters without abbreviations.
Possible values	**TEST**, **PRODUCTION**
Category	Technical details

vads_currency

Description

Numeric currency code to be used for the payment, in compliance with the ISO 4217 standard.

To use a currency other than euro (978), you must request the activation of the "currency conversion" option.

Payment in foreign currency with conversion allows Merchants to present price catalogs in different currencies, but without having to manage their finances in currencies different from the ones specified in their contract.
With this option, when the gateway receives the amount in a currency not managed by your MIDs, it makes a conversion to the company's currency based on the daily exchange rate provided by Visa. For more information, please see the **Offering payment in a foreign currency** chapter of *Hosted Payment Page* guide.

Format

n3

Error code

10

Possible values

Currency	ISO 4217 encoding	Number of digits after the decimal point
Australian Dollar (AUD)	036	2
Cambodian Riel (KHR)	116	0
Canadian Dollar (CAD)	124	2
Chinese Yuan (Renminbi) (CNY)	156	1
Czech Crown (CZK)	203	2
Danish Crown (DKK)	208	2
Hong Kong Dollar (HKD)	344	2
Hungarian Forint (HUF)	348	2
Indian Rupee (INR)	356	2
Indonesian Rupiah (IDR)	360	2
Japanese Yen (JPY)	392	0
South Korean Won (KRW)	410	0
Kuwaiti Dinar (KWD)	414	3
Malaysian Ringgit (MYR)	458	2
Mexican Peso (MXN)	484	2
Moroccan Dirham (MAD)	504	2
New Zealand dollar (NZD)	554	2
Norwegian Crown (NOK)	578	2
Philippine Peso (PHP)	608	2
Russian Ruble (RUB)	643	2
Singapore Dollar (SGD)	702	2
South-African Rand (ZAR)	710	2
Swedish Crown (SEK)	752	2
Swiss Franc (CHF)	756	2
Thai Baht (THB)	764	2
Tunisian Dinar (TND)	788	3
Pound Sterling (GBP)	826	2
US Dollar (USD)	840	2
Taiwan New Dollar (TWD)	901	2
Romanian Leu (RON)	946	2
New Turkish Lira (TRY)	949	2
Euro (EUR)	978	2

Currency	ISO 4217 encoding	Number of digits after the decimal point
Polish Zloty (PLN)	985	2
Brazilian Real (BRL)	986	2

Category Transaction details.

vads_cust_address

Description Buyer's postal address.

Depending on the payment method, certain restrictions can change the format. Please see the technical documentation specific to the payment method for more details.

For Oney payments, the field is mandatory and the format is ans..128.

Only the following special characters are authorized:

- space
- slash (/)
- dash (-)
- apostrophe (')
- comma (,)
- dot (.)

For SEPA Direct Debit payments, the field becomes mandatory if the client has a bank account in the following regions, territories or countries: Switzerland, Monaco, San Marino, Mayotte, St. Pierre and Miquelon, Guernsey, Jersey, Isle of Man.

For Franfinance payments, only the following special characters are authorized: **" ' ` _ + . - @ ,**

Input and output field, returned in the response (IPN and Return URL).

Format ans..255

Note: *The > and < special characters are not authorized.*

Error code 19

Category Buyer details.

vads_cust_address2

Description Address line 2

Input and output field, returned in the response (IPN and Return URL).

Format

Error code	19
Category	Buyer details.

vads_cust_address_number

Description	Street number. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..64
Error code	112
Category	Buyer details.

vads_cust_cell_phone

Description

Buyer's cell phone number.

Accepts all formats:

Examples:

- 0623456789
- +33623456789
- 0033623456789
- (+34) 824 65 43 21
- 87 77 12 34

Depending on the payment method, certain restrictions can change the format. Please see the technical documentation specific to the payment method for more details.

For Oney payments, the field is mandatory and the format is n10.

Input and output field, returned in the response (IPN and Return URL).

Format	an..32
Error code	77
Category	Buyer details.

vads_cust_city

Description

Buyer's city.

Depending on the payment method, certain restrictions can change the format. Please see the technical documentation specific to the payment method for more details.

For Oney payments, the field is mandatory and the format is ans..128. Only the following special characters are authorized:

- space

- slash (/)
- dash (-)
- apostrophe (')

Input and output field, returned in the response (IPN and Return URL).

Format an..128

Error code 21

Category Buyer details.

vads_cust_country

Description Allows to specify the country code in compliance with the ISO 3166 standard.

Input and output field, returned in the response (IPN and Return URL).

Format a2

Error code 22

Examples of possible values

Code	Country	Code	Country
AT	Austria	**GP**	Guadeloupe
CI	Ivory Coast	**MQ**	Martinique
DE	Germany	**NC**	New Caledonia
ES	Spain	**PF**	French Polynesia
FR	France	**PM**	St. Pierre and Miquelon
FR	Corsica	**US**	United States of America

Category Buyer details.

vads_cust_district

Description Buyer's district.

Input and output field, returned in the response (IPN and Return URL).

Format ans..127

Error code 113

Category Buyer details.

vads_cust_email

Description Buyer's email address, **required if you want the payment platform to send an email to the buyer**.

> In order for the buyer to receive an email, don't forget to post this parameter in the form when you generate a payment request.

Input and output field, returned in the response (IPN and Return URL).

Format ans..150

Error code 15

Category Buyer details.

vads_cust_first_name

Description Buyer’s first name.

> Depending on the payment method, certain restrictions can change the format. Please see the technical documentation specific to the payment method for more details.

> For Oney payments, the field is mandatory and the format is an..63.

Input and output field, returned in the response (IPN and Return URL).

Format ans..63

Error code 104

Category Buyer details.

vads_cust_id

Description Buyer ID on the merchant side.

> Depending on the payment method, certain restrictions can change the format. Please see the technical documentation specific to the payment method for more details.

> For Oney payments, the field is mandatory and the format is an..8.

Input and output field, returned in the response (IPN and Return URL).

Format	an..63
Error code	16
Category	Buyer details.

vads_cust_last_name

Description	Buyer's last name.

> Depending on the payment method, certain restrictions can change the format. Please see the technical documentation specific to the payment method for more details.

> For Oney payments, the field is mandatory and the format is an..63.

Input and output field, returned in the response (IPN and Return URL).

Format	ans..63
Error code	105
Category	Buyer details.

vads_cust_legal_name

Description	Buyer's legal name. *Input field.*
Format	ans..100
Error code	121
Category	Buyer details.

vads_cust_name

Description	Buyer's name. This field is deprecated. It is replaced by the **vads_cust_first_name** and **vads_cust_last_name** fields. *Input and output field, returned in the response (IPN and Return URL).*
Format	an..127
Error code	18
Category	Buyer details.

vads_cust_national_id

Description	National identifier. Allows each citizen to identify him/herself with a unique ID within a country. For example, in Brazil, ClearSale requires this field to be populated with the CPF/ CPNJ (in numeric format, between 11 and 20 digits long). *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..255
Error code	124
Category	Buyer details.

vads_cust_phone

Description	Buyer’s phone number. Accepts all formats: Examples: • 0123456789 • +33123456789 • 0033123456789 • (00.571) 638.14.00 • 40 41 42 42

Depending on the payment method, certain restrictions can change the format. Please see the technical documentation specific to the payment method for more details.

For Oney payments, the field is mandatory and the format is n10.

Input and output field, returned in the response (IPN and Return URL).

Format	an..32
Error code	23
Category	Buyer details.

vads_cust_state

Description	Buyer's state/region. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..127
Error code	88
Category	Buyer details.

vads_cust_status

Description	Buyer type. *Input and output field, returned in the response (IPN and Return URL).*
Format	enum
Error code	92
Possible values	**PRIVATE**, **COMPANY**
Category	Buyer details.

vads_cust_title

Description	Buyer's title (e.g. Mr, Mrs, Ms). *Input and output field, returned in the response (IPN and Return URL).*
Format	an..63
Error code	17
Category	Buyer details.

vads_cust_zip

Description	Buyer’s postal code. Depending on the payment method, certain restrictions can change the format. Please see the technical documentation specific to the payment method for more details. For Oney payments, the field is mandatory and the format is n5. *Input and output field, returned in the response (IPN and Return URL).*
Format	an..64
Error code	20
Category	Buyer details.

vads_dcc_amount

Description	Transaction amount expressed in the smallest currency unit defined by **vads_dcc_currency** (cents for euro). This field is populated only if the card is eligible for the Dynamic Currency Conversion (DCC) service and the Buyer has chosen to pay in the currency of the card. The conversion rate is returned in the **vads_change_rate** field. *Output field, returned in the response (IPN and Return URL).*
Format	n..12

Category Transaction details.

vads_dcc_currency

Description

Numeric code in accordance with the ISO 4217 standard of the currency chosen by the Buyer if the card is eligible for the Dynamic Currency Conversion (DCC) service.

This field is populated only if the card is eligible for the Dynamic Currency Conversion (DCC) service and the Buyer has chosen to pay in the currency of the card.

The conversion rate is returned in the **vads_change_rate** field.

Output field, returned in the response (IPN and Return URL).

Format n3

Possible values

Currency	ISO 4217 encoding	Number of digits after the decimal point
Australian Dollar (AUD)	036	2
Cambodian Riel (KHR)	116	0
Canadian Dollar (CAD)	124	2
Chinese Yuan (Renminbi) (CNY)	156	1
Czech Crown (CZK)	203	2
Danish Crown (DKK)	208	2
Hong Kong Dollar (HKD)	344	2
Hungarian Forint (HUF)	348	2
Indian Rupee (INR)	356	2
Indonesian Rupiah (IDR)	360	2
Japanese Yen (JPY)	392	0
South Korean Won (KRW)	410	0
Kuwaiti Dinar (KWD)	414	3
Malaysian Ringgit (MYR)	458	2
Mexican Peso (MXN)	484	2
Moroccan Dirham (MAD)	504	2
New Zealand dollar (NZD)	554	2
Norwegian Crown (NOK)	578	2
Philippine Peso (PHP)	608	2
Russian Ruble (RUB)	643	2
Singapore Dollar (SGD)	702	2
South-African Rand (ZAR)	710	2
Swedish Crown (SEK)	752	2
Swiss Franc (CHF)	756	2
Thai Baht (THB)	764	2
Tunisian Dinar (TND)	788	3
Pound Sterling (GBP)	826	2
US Dollar (USD)	840	2
Taiwan New Dollar (TWD)	901	2
Romanian Leu (RON)	946	2
New Turkish Lira (TRY)	949	2
Euro (EUR)	978	2
Polish Zloty (PLN)	985	2

Currency	ISO 4217 encoding	Number of digits after the decimal point
Brazilian Real (BRL)	986	2

Category Transaction details.

vads_dcc_markup

Description Indicates the percentage (or rate) of sales margin on the total amount.

This field is populated only if the card is eligible for the Dynamic Currency Conversion (DCC) service and the Buyer has chosen to pay in the currency of the card.

Output field, returned in the response (IPN and Return URL).

Format Decimal.

Example: "3,5%" for a margin rate of 3,5%.

Category Transaction details

vads_dcc_rate_date

Description UTC date and time, in YYYYMMDDHHMMSS format, of obtaining the conversion rate from the Dynamic Currency Conversion service (DCC).

This field is populated only if the card is eligible for the Dynamic Currency Conversion (DCC) service and the Buyer has chosen to pay in the currency of the card.

The conversion rate is returned in the **vads_change_rate** field.

Output field, returned in the response (IPN and Return URL).

Format n14

Category Transaction details.

vads_effective_amount

Description Payment amount in the currency used for the capture in the bank.

Output field, returned in the response (IPN and Return URL).

Format n..12

Examples **Example for a shop with capture in EUR**

Payment of EUR 45,25

Parameters sent in the payment form
• vads_amount = 4525 • vads_currency = 978

Returned parameters

- vads_amount = 4525
- vads_currency = 978
- **vads_effective_amount = 4525**
- vads_effective_currency = 978

Payment of USD 10

Parameters sent in the payment form

- vads_amount = 1000
- vads_currency = 840

Returned parameters

- vads_amount = 1000
- vads_currency = 840
- vads_change_rate= 1.0490000000 (rate used)
- **vads_effective_amount = 953 (vads_amount / vads_change_rate)**
- vads_effective_currency = 978

Installment payment of EUR 75.90 in 3 installments

Parameters sent in the payment form

- vads_amount = 7590
- vads_currency = 978
- **vads_payment_config=MULTI_EXT:date1=2590;date2=2500;date3=2500**

  Note: the MULTI_EXT value is not available for SEPA payment.

Returned parameters for the first installment

- vads_amount = 7590
- vads_currency = 978
- **vads_effective_amount = 2590**

Payment of USD 90 paid in 3 installments

Parameters sent in the payment form

- vads_amount = 9000
- vads_currency = 840
- **vads_payment_config=MULTI_EXT:20121025=3000;20121026=2000;20121027=4000**

  Note: the MULTI_EXT value is not available for SEPA payment.

Returned parameters for the first installment

- vads_amount = 9000
- vads_currency = 840
- vads_change_rate= 1.3118 (exchange rate)
- vads_effective_amount = 2287 (**amount of the 1st installment, $30 / vads_change_rate**)

Category	Transaction details.

vads_effective_creation_date

Description	Date of transaction registration in UTC format (GMT+0, 24H) (YYYYMMDDHHMMSS). *Output field, returned in the response (IPN and Return URL).*
Format	n14
Category	Transaction details

vads_effective_currency

Description	Code of the currency used for the capture. *Output field, returned in the response (IPN and Return URL).*
Format	n3
Category	Transaction details

vads_expiry_month

Description	Expiry month of the card used for the payment. *Output field, returned in the response (IPN and Return URL).*
Format	n..2
Category	Payment method details.

vads_expiry_year

Description	Expiry year of the card used for the payment. *Output field, returned in the response (IPN and Return URL).*
Format	n4
Category	Payment method details.

vads_ext_info

Description	Allows to add an optional field to the confirmation e-mail sent to the Merchant. It can be viewed: • In the Merchant Back Office in the transaction details section (**Extras** tab). • In the data transmitted to the merchant website when returning to the shop. • In the data transmitted to the merchant website during the IPN. • By default, in the payment confirmation e-mail sent to the Merchant. • In the payment confirmation e-mail sent to the Buyer, if you specify it in the notification. Required syntax: **vads_ext_info_***fieldname*=value *Input and output field, returned in the response (IPN and Return URL).*
Format	ans
Error code	91
Category	Order details.

vads_ext_info_bil_address_complement

Description	Specific to Brazil and to the ClearSale fraud analyzer. Allows to specify additional information about the billing address. *Input field.*
Format	ans..250
Category	Buyer details.

vads_ext_info_bil_date_of_birth

Description	Allows to transmit the birth date indicated on the bill to the risk analyzer. Format: yyyymmdd *Input field.*
Format	n8
Category	Buyer details.

vads_ext_info_bil_gender

Description	Specific to Brazil and to the ClearSale fraud analyzer. Allows to specify on the receipt whether the buyer is male or female. *Input field.*
Format	n1
Category	Buyer details.

vads_ext_info_deadline

Description	Specific to Brazil and to the ClearSale fraud analyzer. Allows to specify the delivery delay in days (N days). *Input field.*
Format	n
Category	Shipping details.

vads_ext_info_description

Description	Description of the invoice, defined by the merchant during the invoice order creation. *Output field, returned in the response (IPN and Return URL).*
Format	ans..128
Category	Order details.

vads_ext_info_fingerprint_id

Description	Specific to Brazil and to the ClearSale fraud analyzer. Unique session identifier. • This identifier can be generated by the payment gateway. In this case, this parameter must not be populated. • The identifier may also 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.

Input field.

Format string

It is encoded as 128 bytes and can contain uppercase or lowercase characters, numbers or hyphens ([A-Z] [a-z], 0-9, _, -).

Category Buyer details.

vads_ext_info_ship_address_complement

Description Specific to Brazil and to the ClearSale fraud analyzer.

Allows to specify additional information about the shipping address.

Input field.

Format ans..250

Category Shipping details.

vads_ext_info_ship_date_of_birth

Description	Allows to specify the buyer's date of birth for the shipping. Format: yyyymmdd *Input field.*
Format	n8
Category	Shipping details.

vads_ext_info_ship_gender

Description	Specific to Brazil and to the ClearSale fraud analyzer. Allows to specify for the shipping whether the buyer is male or female. *Input field.*
Format	n1
Category	Shipping details.

vads_ext_info_soft_descriptor

Description	Allows to customize the brand name. *Output field, returned in the response (IPN and Return URL).*
Format	ans..255
Category	Order details

vads_ext_trans_id

Description	External transaction reference. Example: Klarna reservation number, PayPal reservation number. *Output field, returned in the response (IPN and Return URL).*
Format	ans..20
Category	Transaction details

vads_extra_result

Description

Risk assessment result (specific to shops that have enabled the feature). Its meaning depends on the value entered in **vads_result**.

- If **vads_result** equals 30 (request error), then **vads_extra_result** contains the numeric code of the field with an error in the value or the format. This value can be set to 99 in case of an unknown error in the request.

  Example: if **vads_extra_result** contains the value 09, it means that the amount specified in **vads_amount** is incorrect (for example, if the amount contains decimals, as it would not have been converted to cents in advance).

- If **vads_result** equals 05 (declined) or 00 (accepted), **vads_extra_result** contains the numeric code of the risk management result.

Code	Description
Empty	No verification completed.
00	All the verification processes have been successfully completed.
02	Credit card velocity exceeded.
03	The card is on the Merchant's greylist.
04	The country of origin of the card is on the Merchant's greylist.
05	The IP address is on the Merchant's greylist.
06	The BIN code is on the Merchant's greylist.
07	Detection of an e-carte bleue.
08	Detection of a national commercial card.
09	Detection of a foreign commercial card.
14	Detection of a card that requires systematic authorization.
20	Relevance verification: countries do not match (country IP address, card country, buyer's country).
30	The country of the this IP address is on the greylist.
99	Technical issue encountered by the server during a local verification process.

Output field, returned in the response (IPN and Return URL).

Category Technical information.

vads_first_installment_delay

Description

When the acquirer supports the parameter, this field allows to specify the number of deferred months to be applied on the first due date of the payment in installments (e.g.: Webpay Completa).

The payment will be declined and the **vads_payment_error** field will be valued at 171 in the following cases:

- the merchant is not allowed to defer payments,
- the value transmitted in the request is not among the options authorized by the acquirer.

Input field.

Format n..2

Error code N/A

Category Transaction details

vads_hash

Description

A unique key returned only to the Instant Payment Notification (IPN).

Output field, returned in the response (IPN only).

Format

an64

Category

Technical information.

vads_identifier

Description

Unique identifier (token or UMR) associated with a payment method.

- This identifier can 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.

Input and output field, returned in the response (IPN and Return URL).

Format

Two possible formats:

- **an32**: if the identifier is generated by the payment gateway. This format is only used by the payment gateway.
- **ans..50**: if the identifier is generated by the merchant. Cannot be an32 in this case.

Error code

30

Category

Recurring payment details.

vads_identifier_previously_registered

Description

When the "Check token uniqueness" option is enabled, every time a token is created the gateway checks the uniqueness of the payment method. If the payment method has already been registered (same number and expiration date), then the **vads_identifier_previously_registered** field is transmitted with the **true** value in the end of payment notification.

The **vads_identifier_previously_registered** field is never returned in the end of payment notification if no duplicate payment methods are detected. Therefore, the **false** value is never sent to the merchant website.

Output field, returned in the response (IPN only).

Format

bool

Category

Information about the subscription.

vads_identifier_status

Description

Only present if the requested action is token creation or update (or UMR for SEPA payment).

Output field, returned in the response (IPN and Return URL).

Format

string

Possible values

Value	Description
CREATED	The authorization request has been accepted. The token (or UMR for SEPA payment) has been successfully created.
NOT_CREATED	The authorization request has been declined. The token (or UMR for SEPA payment) has not been created, and therefore cannot be viewed in the Merchant Back Office.
UPDATED	The token (or UMR for SEPA payment) has been successfully updated.
NOT_UPDATED	The token (or UMR for SEPA payment) has not been updated.
ABANDONED	The action has been abandoned by the buyer (debtor). The token (or UMR for SEPA payment) has not been created, and therefore cannot be viewed in the Merchant Back Office.

Category

Information about the subscription.

vads_iframe_options

Description

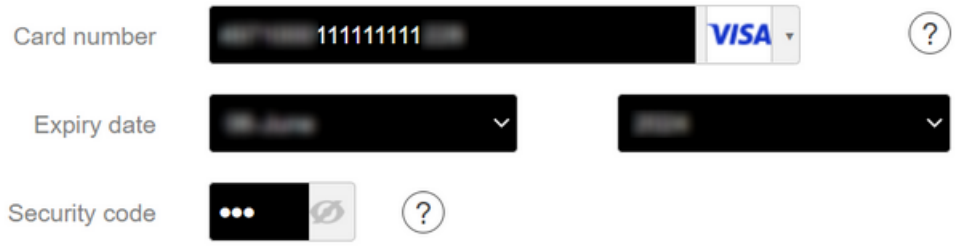
Allows to customize certain elements on the payment page in iframe mode:

- **fieldsBackgroundColor** : background color of entry fields
- **fieldsFontColor** : font color in entry fields

Example of syntax:

```
vads_iframe_options =
 {"fieldsBackgroundColor":"#000000","fieldsFontColor":"#FFFFFF"}
```

The result will be:



Input field.

Format

json

Error code

In case of a format error, the field is ignored and the payment form is not rejected.

Category

Payment page customization

vads_initial_issuer_transaction_identifier

Description

Unique transaction reference generated by the issuer.

Called a "chaining reference", it is used under PSD2 for processing **MIT** transactions to indicate to the issuer that the transaction is part of a series of payments for which the cardholder authenticated himself in the first payment.

Without this information, the issuer can refuse an **MIT** transaction for lack of authentication (soft decline).

Output field, returned in the response (IPN and Return URL).

Format ans32

Examples 103E58BCD2311680 | 2MCSYFB0QD0126 | 1RCB202633021958

Category Transaction details.

vads_insurance_amount

Description

Insurance amount for the entire order.

Concerns only the PayPal payment method.

Input field.

Format n..12

Error code 110

Category Order details.

vads_language

Description

In the payment request:

Defines the language of the payment page (ISO 639-1 standard).

If the field has not been sent or is empty, the payment page will be shown in the language of the buyer's browser.

In the response:

Returns the value specified in the form if the buyer has not changed the language of the payment page.

Returns the language selected by the buyer if the buyer has changed it by clicking on a different flag.

Input and output field, returned in the response (IPN and Return URL).

Format a2

Error code 12

Possible values

Language	ISO 639-1 standard
German	de
English	en
Chinese	zh
Spanish	es

Language	ISO 639-1 standard
French	fr
Italian	it
Japanese	ja
Dutch	nl
Polish	pl
Portuguese	pt
Russian	ru
Swedish	sv
Turkish	tr

Category Payment page customization.

vads_nb_products

Description Allows to define the number of items 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 Merchant 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 is mandatory to populate the other fields that start with **vads_product_**.*

Input field.

Format n..12

Category Order details.

vads_occurrence_type

Description Allows to identify if the transaction is part of a series of payments (subscription or payment in installments).

Useful for accurately identifying the first payment of a series.

With the application of Soft Decline, the payment gateway automatically makes a new payment attempt with 3D Secure authentication, when possible. This changes the payment sequence number. The **vads_sequence_number** field no longer allows to easily identify the first payment of a series.

Output field, returned in the response (IPN and Return URL).

Format enum

Possible values

- **RECURRENT_INITIAL:** First payment of a series.
- **RECURRENT_INTERMEDIAIRE:** Nth payment of a series.
- **RECURRENT_FINAL:** Last payment of a series.
- **UNITAIRE:** Single payment (immediate payment).

Category Transaction details.

vads_operation_type

Description

Allows to identify the type of operation: debit, credit (refund) or verification when creating or updating a token without a transaction, or when requesting information as part of a payment upon shipment.

Output field, returned in the response (IPN and Return URL).

The **vads_operation_type** field is not returned in the response when a payment is canceled or abandoned.

Format enum

Possible values

- **DEBIT**
- **CREDIT**
- **VERIFICATION**

*The **vads_operation_type** field is set to **VERIFICATION** in the following cases where there is no transaction:*

- *vads_page_action = REGISTER*
  Creating a token without payment
- *vads_page_action = REGISTER_UPDATE*
  Updating token details
- *vads_page_action = REGISTER_SUBSCRIBE*
  Creating a token during a recurring payment

Category Transaction details.

vads_order_description

Description

Order description.

String of characters transmitted by the merchant, visible on the payment receipt and the payment confirmation e-mail sent to the buyer.

Used in Colombia for all payment methods.

Input and output field, returned in the response (IPN and Return URL).

Format ans..65

Error code 173

Category Order details.

vads_order_id

Description

Order ID. It is also included in the payment confirmation e-mail sent to the buyer.

The maximum field size is 64 characters.

Accepted characters are:

- a b c d e f g h i j k l m n o p q r s t u v w x y z
- A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
- 0 1 2 3 4 5 6 7 8 9
- _-

Depending on the payment method, certain restrictions can change the format. Please see the technical documentation specific to the payment method for more details.

Acquirer network	Length	Accepted characters
ACCORD	9	• a b c d e f g h i j k l m n o p q r s t u v w x y z • A B C D E F G H I J K L M N O P Q R S T U V W X Y Z • 0 1 2 3 4 5 6 7 8 9
ACCORD_SANDBOX	9	• a b c d e f g h i j k l m n o p q r s t u v w x y z • A B C D E F G H I J K L M N O P Q R S T U V W X Y Z • 0 1 2 3 4 5 6 7 8 9
ANCV	1..20	• a b c d e f g h i j k l m n o p q r s t u v w x y z • A B C D E F G H I J K L M N O P Q R S T U V W X Y Z • 0 1 2 3 4 5 6 7 8 9 • _-
GIROPAY	1..27	• a b c d e f g h i j k l m n o p q r s t u v w x y z • A B C D E F G H I J K L M N O P Q R S T U V W X Y Z • 0 1 2 3 4 5 6 7 8 9 • _-
KLARNA	1..32	• a b c d e f g h i j k l m n o p q r s t u v w x y z • A B C D E F G H I J K L M N O P Q R S T U V W X Y Z • 0 1 2 3 4 5 6 7 8 9
PAYDIREKT_V2	1..20	• a b c d e f g h i j k l m n o p q r s t u v w x y z • A B C D E F G H I J K L M N O P Q R S T U V W X Y Z • 0 1 2 3 4 5 6 7 8 9 • _-

Input and output field, returned in the response (IPN and Return URL).

Format ans..64

Error code 13

Category Order details.

vads_order_info

Description	Order description. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..255
Error code	14
Category	Order details.

vads_order_info2

Description	Order description. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..255
Error code	14
Category	Order details.

vads_order_info3

Description	Order description. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..255
Error code	14
Category	Order details.

vads_override_payment_cinematic

Description	Optional parameter. Used by the merchant to request, on individual transactions, a payment workflow different from the one specified in his or her contract ("Payment workflow" field). *Input field.* ***Note*** *Only certain contracts exploit this parameter. If a value is selected in a contract that does not exploit the parameter, the data is ignored and no error message is raised.*
Format	enum
Error code	131
Possible values	• (empty) The MID value is used. • IMMEDIATE_CAPTURE Corresponds to a workflow of immediate capture: the capture is triggered by the acquirer on the day of the transaction. • DELAYED_CAPTURE

Corresponds to a workflow of deferred capture: the capture is triggered by the payment gateway, always before the expiry of the authorization request.

Category	Technical details

vads_page_action

Description

Mandatory parameter.

Defines the action to be performed.

Input and output field, returned in the response (IPN and Return URL).

Format

enum

Error code

46

Possible values

- PAYMENT
  Payment (using token or not)
- REGISTER
  Creation of a token without payment
- REGISTER_UPDATE
  Update of information associated with the token
- REGISTER_PAY
  Creation of a token during a payment
- REGISTER_SUBSCRIBE
  Creation of a token during a recurring payment
- REGISTER_PAY_SUBSCRIBE
  Creation of a token during creation of a subscription with payment
- SUBSCRIBE
  Using a token to create a recurring payment
- REGISTER_UPDATE_PAY
  Update of information associated with the token during a payment
- ASK_REGISTER_PAY
  Payment with option for the cardholder to create a token

Category

Technical details

vads_payment_cards

Description

Contains the list of payment methods offered to the buyer, separated by “;”.

Example: "VISA;MASTERCARD".

If this list contains one payment method, the detail entry page for this payment method will directly appear. If not, the payment method selection page is displayed.

All eligible payment methods (currencies, technical constraints, etc.) associated with the shop will be displayed if vads_payment_cards is not sent or if it is sent empty.

Input field.

Format

type 1;type 2;type 3

Error code

08

The form has been rejected in the following cases:

- The transmitted value does not appear on the list below.
- The field is populated with TOUTES, ALL.

  To offer all payment methods, this parameter should not be posted or be posted empty.
- The transmitted value does not correspond to the payment method available for your shop.
- Your e-commerce contract has been closed by your banking institution.
- The transmitted value is not eligible in the associated network.

Possible values

Payment method	Card types (vads_payment_cards)
American Express	AMEX
Apetiz electronic meal voucher	APETIZ
Cpay card	AURORE-MULTI
CB	CB
Chèque Déjeuner electronic meal voucher	CHQ_DEJ
Conecs electronic meal voucher	CONECS
Chèque-Vacances Connect	CVCO
e-carte bleue	E-CARTEBLEUE
Franfinance payment in 3X	FRANFINANCE_3X
Franfinance payment in 4X	FRANFINANCE_4X
Payment in 3 installments with no fees with BNPP PF	FULLCB3X
Payment in 4 installments with no fees with BNPP PF	FULLCB4X
Illicado Gift Card	ILLICADO
JouéClub gift card - Sandbox mode	ILLICADO_SB
Maestro	MAESTRO
Mastercard	MASTERCARD
MasterPass	MASTERPASS
Paiement 3x 4x Oney	ONEY_3X_4X
Payment 10x 12x Oney	ONEY_10X_12X
Payment Oney Pay Later	ONEY_PAYLATER
Cartes Enseignes partenaires d'Oney.	ONEY_ENSEIGNE
PayPal	PAYPAL
PayPal - Sandbox mode	PAYPAL_SB
SEPA DIRECT DEBIT	SDD
Sodexo electronic meal voucher	SODEXO

Payment method	Card types (vads_payment_cards)
Visa	VISA
Visa Electron	VISA_ELECTRON
Vpay	VPAY

Category Transaction details.

vads_payment_certificate

Description This field is populated by the payment gateway if the authorization has been **successfully** completed.

Output field, returned in the response (IPN and Return URL).

Format an40

Category Transaction details

vads_payment_config

Description

Defines the type of payment: immediate or installment.

- For a single payment, the value must be set to **SINGLE**.
- For an installment payment with fixed amounts and dates, the value must be set to **MULTI:** followed by key=value pairs separated by the "**;**" character.

  The parameters are:

  - "**first**" indicates the amount of the first installment (populated in the smallest unit of the currency).
  - "**count**" indicates the total number of installments.
  - "**period**" indicates the number of days between 2 installments.

    The field order associated with MULTI must be respected.

    The activation of the payment in installments feature is subject to the prior agreement of Société Générale.

    Note: the MULTI value is not available for SEPA payment.
- For an installment payment with a customized installment schedule, the value must be set to **MULTI_EXT:** followed by the date=amount pairs separated by the "**;**" character.

  The dates must not be passed.

  Note: the MULTI_EXT value is not available for SEPA payment.

Using the MULTI_EXT value requires a subscription to the **Advanced installment payment** option.

Note: *The value of **vads_capture_delay** is not taken into account in the case of payment in installments **MULTI_EXT**.*

Input and output field, returned in the response (IPN and Return URL).

Format

enum

Error code

07

Possible values

- **SINGLE**
- **MULTI:first= initial_amount;count=installments_nb ;period=interval_in_days**

  Note: the MULTI value is not available for SEPA payment.
- **MULTI_EXT:date1=amount1;date2=amount2;date3=amount3**

  Note: the MULTI_EXT value is not available for SEPA payment.

Example 1

MULTI allows to define an installment payment.

The amount of each installment corresponds to the total amount divided by the number of installments.

The amount of the first installment can be different, it can be specified in **first** parameter.

In case the remaining amount does not equal zero, it will be added up to the amount of the last installment.

Payment request:

- vads_capture_delay=2
- vads_currency=978
- vads_amount=20000
- vads_payment_config=MULTI:first=10000;count=4;period=30

Result:

A first payment of EUR 100.00 will be captured by the bank in 2 days (vads_capture_delay).

A second payment of EUR 33.33 will be made in 32 days (vads_capture_delay + period).

A third payment of EUR 33.33 will be made in 62 days.

A fourth payment of EUR 33.34 will be made in 92 days.

The total amount is EUR 200.00 (vads_amount= 20000). The remaining amount has been added to the amount of the last installment.

This instruction allows to immediately create 4 payments with the same transaction number but different sequence numbers (vads_sequence_number).

Example 2

MULTI_EXT allows to define a customized installment schedule. You will be able to define the amount of each installment.

MULTI_EXT : payment request:

- vads_currency=978
- vads_amount=19050
- vads_payment_config= MULTI_EXT**:**20150601 =10000; 20150701 =4525; 20150808 =4525

Result:

The first payment of EUR 100.00 is scheduled for June 1st 2015.

The second payment of EUR 45.25 is scheduled for July 1st 2015.

The last payment of EUR 45.25 is scheduled for August 8th 2015.

Note:

The total amount must be equal to the value of the **vads_amount** field. The date of the last installment cannot be later than 12 months after the date of submission of the form. If the last installment is scheduled later than the card expiry date, no installment will be registered and the buyer will be notified about this issue.

Category

Transaction details.

vads_payment_error

Description Error codes that may appear when a payment has been declined.

Output field, returned in the response (IPN and Return URL).

Format n..3

Possible values

Code	Message
1	Transaction not found.
2	Transaction not found.
3	This action has not been authorized for a transaction with the X status.
4	This transaction is not authorized in this context.
5	This transaction already exists.
6	Invalid transaction amount.
7	This operation is no longer allowed for a transaction created on this date.
8	The payment method exp. date does not allow to process this action.
9	Required security code.
10	The credit amount is higher than the initial amount.
11	The credit amount is higher than the initial amount.
12	Credit duplication (refund) is not authorized.
13	A technical problem occurred. We are not able to process your request.
14	A technical problem occurred. We are not able to process your request.
15	A technical problem occurred. We are not able to process your request.
16	A technical problem occurred. We are not able to process your request.
19	Unknown currency.
20	Invalid payment method.
21	No Merchant ID found for this payment. Please modify the data or contact your sales contact in case of repeated failures.
22	Shop not found.
23	Merchant ID (MID) unclear.
24	Merchant ID (MID) invalid.
25	A technical problem occurred. We are not able to process your request.
26	Invalid card number
27	Invalid card number.
28	Invalid card number.
29	Invalid card number.
30	Invalid card number (Luhn).
31	Invalid card number (length).
32	The card number does not match the selected payment method.
33	The card number does not match the selected payment method.
34	Card with unconditional authorization control failed.
35	E-carte bleue control failed.
36	The transaction has been refused by risk management.
37	Interruption not processed during the payment.
38	A technical problem occurred. We are not able to process your request.
39	3D Secure was declined for this transaction
40	A technical problem occurred. We are not able to process your request.
41	A technical problem occurred. We are not able to process your request.
42	An internal problem occurred during the card number checking.
43	An internal problem occurred during the card number checking.
44	Unauthorized action for face-to-face transactions.
45	Invalid currency for this change.

Code	Message
46	The amount exceeds the maximum authorized amount.
47	The requested capture date is later than the authorization validity date.
48	The required change is not valid.
49	Invalid definition of installment payment.
50	Unknown POS.
51	Unknown exchange rate.
52	This Merchant ID (MID) has been closed since aaaa/mm/dd.
53	The TEST shop has been closed since aaaa/mm/dd.
54	Rejected parameter that may contain sensitive data.
55	A technical problem occurred. We are not able to process your request.
56	The The amount is lower than the authorized minimum amount.
57	Error retrieving the alias.
58	The alias status is not compatible with this operation.
59	Error retrieving the alias.
60	This token already exists.
61	Invalid token.
62	Token creation failed.
63	This recurring payment already exists.
64	This recurring payment is already terminated.
65	Invalid recurring payment.
66	The rule of recurring payment is not valid.
67	Creation of the recurring payment declined.
68	Cancellation rejected.
69	A technical problem occurred. We are not able to process your request.
70	Invalid country code.
71	Invalid web service parameter.
72	Authorization declined by Cofinoga.
73	Authorization declined for EUR 1 (or information request about the CB network if the acquirer supports it).
74	Invalid payment configuration.
75	The operation was declined by PayPal.
76	The cardholder’s name is absent.
77	A technical problem occurred. We are not able to process your request.
78	Transaction ID missing.
79	This transaction ID is already used.
80	Transaction ID expired.
81	The content of the configuration theme is not valid.
82	Refund is not authorized for this Merchant ID (MID).
83	Transaction amount outside the allowed values.
84	Capture not authorized for transaction X with the order number XX because as it is not yet registered in a CNAB/Remessa file.
85	Commission absent upon boleto capture.
86	Capture(s) not authorized for transaction(s) X as it is not yet registered in a CNAB/Remessa file.
87	A technical problem occurred. We are not able to process your request.
88	Refund error: PayPal does not allow transaction refunds after 60 days.
89	The modification is not authorized.
90	An error occurred during the refund of this transaction.
91	No payment options have been enabled for this MID.
92	An error occurred while calculating the payment channel.
93	An error occurred during buyer redirection to the page of payment finalization.
94	A technical error occurred during the call to the RSP service.
96	An error occurred during the capture of this transaction.
97	The requested capture date is too far.
98	Invalid transaction date.

Code	Message
99	An error occurred while calculating the payment source.
100	Failed commercial card verification.
101	Rejected as the first installment has been rejected.
103	The transaction status could not be synchronized with the external system.
104	An error occurred during the capture of this transaction.
105	3D Secure - Invalid signature of the authentication message (Pares).
106	Unsupported currency on this Merchant ID (MID) and/or shop.
107	The payment method associated with the token is no longer valid.
108	A technical problem occurred. We are not able to process your request.
109	Timeout during buyer redirection.
110	Payment method not supported by the Merchant ID (MID).
111	Refusal of transactions without Liability shift.
112	Cancellation is not authorized.
113	Duplication is not authorized.
115	Refund is not authorized.
116	Manual payment not authorized for this payment method.
118	Payment in installments not authorized for this payment method.
119	The submitted date is invalid.
120	The initial transaction option is not applicable.
124	Inactive payment method.
125	Payment refused by the acquirer.
126	This action is not possible because the sequence of payment is not completed.
128	Invalid payment method.
129	Invalid PIN code.
130	Out of credit.
131	Insufficient balance.
136	The derivative transactions have been refused without for the initial transaction.
137	Duplicate transaction.
138	Partial refund is impossible for this transaction.
139	Refund rejected.
140	Due to a technical problem, we are unable to process your request.
141	The risk analyzer rejected this transaction.
142	The used payment method is not valid for the requested payment mode.
143	A technical problem occurred. We are not able to process your request.
144	A transaction in production mode has been marked as in test mode by the acquirer.
145	A transaction in test mode has been marked as in production mode by the acquirer.
146	Invalid SMS code.
147	The risk assessment module asked for this transaction refusal.
148	No compatible MIDs found.
149	The payment session has expired (the Buyer has been redirected to the ACS and has not finalized the 3D Secure authentication).
150	No compatible MIDs found.
151	A Facily Pay transaction cannot be canceled/modified/refunded between 11.30 p.m. and 5.30 a.m.
152	A technical problem occurred. We are not able to process your request.
153	A technical error occurred during the call to the Banque Accord service.
155	The Facily Pay transaction could not be canceled/edited/refunded: the transaction status does not allow to perform the requested action. Reminder regarding a Facily Pay transaction: a refund must be made within two days after the capture, the delay between two refunds is one day, a partial refund is limited to 20 days, a full refund is limited to 6 months.
156	Operation not supported.
158	A technical problem occurred. We are not able to process your request.
159	The amount is less than the minimum amount authorized (minimum= X).
160	It is impossible to refund the transaction X as it has been subject to chargeback.

Code	Message
161	The modification failed because the chosen payment option is not available.
162	The modification failed because the chosen payment option is no longer valid.
163	The modification failed because the chosen payment option does not exist.
164	Invalid payment option.
165	The ID type is present, but its number is absent.
166	The ID number is present, but its type is absent.
167	The ID type is unknown.
168	The ID number is invalid.
169	The specific data that must be transmitted to the acquirer is invalid.
170	Deferred payment is not authorized.
171	The number of months for the deferred payment is not authorized.
172	The selected payment process is invalid.
173	Error within the Express Checkout PayPal service.
174	Card issuer unavailable.
175	Cancellation impossible, please try a refund.
176	Refund impossible, please try a cancellation.
177	No response to the authorization request was received within the fixed time-frame.
178	Cancellation impossible, the transaction has already been canceled.
179	The transaction status is unknown.
182	The customer's national identifier is absent.
183	The format of the customer's national identifier is incorrect.
184	The e-mail is absent.
186	The minimum authorized amount cannot make up less than 80% of the initial amount.
187	In order to refund the transaction, please contact RBM at solicitudes@rbm.com.co.
188	In order to refund the transaction, please contact Credibanco at atrecom@credibanco.com.
189	In order to refund the transaction, please contact Davivienda at wemedellin@davivienda.com.
190	The reason for refusal does not allow transaction duplication.
191	The billing address is absent or incomplete.
192	Manual capture is not allowed for this type of contract.
193	Amplification refused by the issuer. This amplification authorization refusal does not affect the initial authorization, which is still valid.
194	Credit is not allowed for this transaction type.
195	The amount eligible in TRD is invalid.
196	The amount eligible in TRD is negative.
197	The amount eligible in TRD is greater than the order amount.
198	The data transmitted to the CONECS network in the vads_acquirer_transient_data field does not contain the eligibleAmount key.
199	The amount eligible in TRD is lower than €1.50.
200	The specific data that must be transmitted to the acquirer is invalid.
201	The Buyer's name is absent or incomplete.
202	Payment token canceled.
203	Payment method verification rejected.
204	An error occurred during the cancellation of this transaction.
205	3D Secure - cannot reach DS or ACS.
206	3D Secure - A technical error occurred during the process. For more details, see the *Transaction with failed 3D Secure authentication* article via the *Retrieving the cardholder authentication result* chapter.
207	3D Secure - Refusal of the authentication by the issuer.. For more details, see the *Transaction with failed 3D Secure authentication* article via the *Retrieving the cardholder authentication result* chapter.
208	3D Secure - Refusal as authentication by the issuer is impossible.
210	Duplication of verification type transactions forbidden.
211	In order to refund the transaction, please contact Tuya.
212	In order to refund the transaction, please contact BigPass Edenred Colombia at sercliente-co@edenred.com.
213	3D Secure - Session altered by the ACS.

Code	Message
214	The card number is not eligible for this payment.
215	Internal error acquirer on the acquirer's side.
216	Expired OTP code.
217	Invalid OTP code.
218	Invalid data transmitted to the authentication validation service.
219	A technical error occurred during the authentication.
220	An internal error occurred during the authentication.
221	The address is required following the entry of an IBAN outside the EEA zone.
222	The authentication has been canceled.
223	The selected token cannot be used by the Visanet network.
224	Unknown cardholder
225	The data received from the wallet is not consistent.
226	Unable to access the wallet.
227	Authentication impossible.

Category Technical information.

vads_payment_option_code

Description Code of the used payment option.

Input and output field, returned in the response (IPN and Return URL).

Format an..5

Error code 103

Category Transaction details

vads_payment_seq

Description

Details of performed transactions.

Output field, returned in the response (IPN and Return URL).

Format

json

The **vads_payment_seq** field (json format) describes the split payment sequence. It contains:

- **"trans_id"**: transaction identifier used for the entire payment sequence.
- **"transaction"**: transaction table of the sequence. It contains:

Field name	Description
amount	Amount of the payment sequence.
operation_type	Debit transaction.
auth_number	Authorization number. Will not be returned if not applicable to the used payment method. Example: 949478
auth_result	Return code of the authorization request.
capture_delay	Delay before the capture (in days). • For a payment by card, this parameter is the requested capture date (ISO 8601 format). If not sent in the payment form, the value defined in the Merchant Back Office will be used.
card_brand	Used payment method. For a payment by card (e.g. CB or Visa or MasterCard co-branded CB cards), this parameter is set to **"CB"**. See the Payment Gateway Implementation Guide available in our online documentation archive to see the complete list of card types.
card_number	Payment method number.
expiry_month	Expiry month of the payment method.
expiry_year	Expiry year of the payment method.
payment_certificate	Payment certificate.
contract_used	Contract used for the payment.
identifier	Unique identifier (token) associated with a payment method.
identifier_status	Only present if the requested action is token creation or update. Possible values:
presentation_date	For a payments by card, this parameter is the requested capture date (ISO 8601 format).
trans_id	Transaction number.

Value	Description
CREATED	The authorization request has been accepted. The token (or UMR for SEPA payment) has been successfully created.
NOT_CREATED	The authorization request has been declined. The token (or UMR for SEPA payment) has not been created, and therefore cannot be viewed in the Merchant Back Office.
UPDATED	The token (or UMR for SEPA payment) has been successfully updated.
NOT_UPDATED	The token (or UMR for SEPA payment) has not been updated.
ABANDONED	The action has been abandoned by the buyer (debtor). The token (or UMR for SEPA payment) has not been created, and therefore cannot be viewed in the Merchant Back Office.

<table>
<tr><th>Field name</th><th>Description</th></tr>
<tr><td>ext_trans_id</td><td>This field is not sent for credit card payments.</td></tr>
<tr><td>trans_uuid</td><td>Unique reference generated by the payment gateway after the creation of a payment transaction.
Guarantees that each transaction is unique.</td></tr>
<tr><td>extra_result</td><td>Numeric code of the risk assessment result.
<table><tr><th>Code</th><th>Description</th></tr><tr><td>Empty</td><td>No verification completed.</td></tr><tr><td>00</td><td>All the verification processes have been successfully completed.</td></tr><tr><td>02</td><td>Credit card velocity exceeded.</td></tr><tr><td>03</td><td>The card is on the Merchant’s greylist.</td></tr><tr><td>04</td><td>The country of origin of the card is on the Merchant’s greylist.</td></tr><tr><td>05</td><td>The IP address is on the Merchant’s greylist.</td></tr><tr><td>06</td><td>The BIN code is on the Merchant’s greylist.</td></tr><tr><td>07</td><td>Detection of an e-carte bleue.</td></tr><tr><td>08</td><td>Detection of a national commercial card.</td></tr><tr><td>09</td><td>Detection of a foreign commercial card.</td></tr><tr><td>14</td><td>Detection of a card that requires systematic authorization.</td></tr><tr><td>20</td><td>Relevance verification: countries do not match (country IP address, card country, buyer's country).</td></tr><tr><td>30</td><td>The country of the this IP address is on the greylist.</td></tr><tr><td>99</td><td>Technical issue encountered by the server during a local verification process.</td></tr></table></td></tr>
<tr><td>sequence_number</td><td>Sequence number.</td></tr>
<tr><td>trans_status</td><td>Status of the transaction.</td></tr>
</table>

Table 2: JSON object content

Note: canceled transactions also appear in the table (information provided in the JSON trans_status parameter).

Category Transaction details.

vads_payment_src

Description Allows to define the entry mode of payment method details.

Input and output field, returned in the response (IPN and Return URL).

Format enum

Error code 60

Possible values

Value	Description
EC	E-commerce: Data entry on the payment page by the cardholder.
MOTO	MAIL OR TELEPHONE ORDER: Entry made by an operator. Payment method details are transmitted by post or by e-mail.
CC	Call center: Entry made by a call center operator.
OTHER	Another sales channel. Returned output value for payments made via the Merchant Back Office, payments by file, recurring payments, proximity payments, refunds via the Shopify CMS.
Missing or empty	The payment is made in e-commerce (**EC**) mode.

Only the **EC** value allows to create a transaction with 3D Secure.

The other values must only be used for distance sales, where 3D Secure is not applicable.

Category Transaction details.

vads_pays_ip

Description Buyer's country code and the IP address in the ISO 3166 format.

Output field, returned in the response (IPN and Return URL).

Format a2

Category Buyer details.

vads_presentation_date

Description

- Date and time in UTC format of requested capture in the bank, in YYYYMMDDHHMMSS format.

  or

- Date and time in UTC format of an installment requested as part of the SEPA direct debit.

Output field, returned in the response (IPN and Return URL).

Format n14

Category Transaction details.

vads_pretax_amount

Description Allows to define the **pre-tax** amount of the whole order.

The value must be specified in the smallest currency unit (cents for euro).

Output field, returned in the response (IPN and Return URL).

Format n..12

Category Order details.

vads_product_amountN

Description Allows to define the amount of each item in the cart.

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

The amount is expressed in the smallest currency unit (cents for euro).

The payment form will be rejected in the following cases:

- a negative amount [vads_product_amountN=-100],
- an amount with decimals or points [vads_product_amountN=100.50]

Input field.

Format	n..12
Error code	102
Category	Order details.

vads_product_ext_idN

Description	Corresponds to the product barcode on the merchant's website. N corresponds to the reference of the article (0 for the first item, 1 for the second item, etc.). Field transmitted to the Konduto fraud analyzer. *Input field.*
Format	an..100
Error code	120
Category	Order details.

vads_product_labelN

Description	Allows to define the label of each item in the cart. N corresponds to the reference of the article (0 for the first item, 1 for the second item, etc.). Depending on the payment method, certain restrictions can change the format. Please see the technical documentation specific to the payment method for more details. For Oney payments, the field is mandatory and the format is an..127. *Input field.*
Format	ans..255
Error code	97
Category	Order details.

vads_product_qtyN

Description	Allows to define the quantity of each item in the cart. N is an integer that corresponds to the index of the item (0 for the first item, 1 for the second item, etc.). *Input field.*
Format	n..12
Error code	101
Category	Order details.

vads_product_refN

Description	Allows to define the reference of each item in the cart. N corresponds to the reference of the article (0 for the first item, 1 for the second item, etc.). *Input field.*
Format	an..64
Error code	100
Category	Order details.

vads_product_typeN

Description Allows to define the type of each item in the cart.

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

Input field.

Format enum

Error code 98

Possible values

Value	Description
FOOD_AND_GROCERY	Food and grocery
AUTOMOTIVE	Cars / Moto
ENTERTAINMENT	Entertainment / Culture
HOME_AND_GARDEN	Home / Gardening
HOME_APPLIANCE	Household appliances
AUCTION_AND_GROUP_BUYING	Auctions / Group purchasing
FLOWERS_AND_GIFTS	Flowers / Presents
COMPUTER_AND_SOFTWARE	Computers / Software
HEALTH_AND_BEAUTY	Health / Beauty
SERVICE_FOR_INDIVIDUAL	Services for individuals
SERVICE_FOR_BUSINESS	Services for companies
SPORTS	Sports
CLOTHING_AND_ACCESSORIES	Clothes / Accessories
TRAVEL	Travel
HOME_AUDIO_PHOTO_VIDEO	Audio / Photo / Video
TELEPHONY	Telephony

Category Transaction details.

vads_product_vatN

Description Allows to define the tax for each item in the cart.

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

Input field.

Format n..12

Error code 203

Possible values

- **An integer without a decimal separator**

  To display an amount in cents applied to the product in question.

  E.g.: 4525 for EUR 45.25

- **A decimal number lower than 100**

  To display a percentage applied to the payment amount for the product in question with maximum 4 digits after the decimal point.

  Examples: 20.0 or 19.6532

  Notes:

- *The decimal separator is mandatory for displaying a percentage.*
- *The decimal separator is represented by the "." symbol.*

Category Order details.

vads_proof_of_id_number

Description Field reserved to the entry of the buyer's ID number on the payment page.

The format depends on the ID type and allows 7 to 13 characters, digits, letters and/or points.

In Latin America, this parameter is required for DECIDIR.

Input field.

Format an..13

Error code 129

Category Buyer details.

vads_proof_of_id_type

Description Field reserved for Latin America.

This field allows to pre-fill the buyer's ID type that is required for the payment.

The value to transmit depends on the acquirer.

Input field.

Format enum

Possible values

Country	Value	Description
Argentina	**DNI**	Documento Nacional de Identidad
Brasil	**CNPJ**	Cadastro Nacional da Pessoa Jurídica
	CPF	Cadastro de Pessoas Físicas
Colombia	**CC**	Cédula de Ciudadania
	TI	Tarjeta de Identidad
	CE	Cédula de Extranjeria
	NI	Número de Identificación Tributaria
	PS	Pasaporte
	RN	Registro Civil de Nacimiento
	DE	Documenta de Identificacion Extranjero
	TE	Tarjeta de Extranjeria
Peru	**DNI_PER**	Documento National de Identidad
	PAR	Partida de Nacimiento
	PAS	Pasaporte
	LMI	Libreta Militar
	NAN	Otro

Error code 128

Category Buyer details.

vads_recurrence_number

Description Recurrence number of the subscription

Output field, returned in the response (IPN and Return URL).

Format n..2

Category Subscription details.

vads_recurrence_status

Description Recurring payment status.

Appears only if the requested action concerns creating or updating a recurring payment (REGISTER_SUBSCRIBE, SUBSCRIBE, REGISTER_PAY_SUBSCRIBE, REGISTER_UPDATE_PAY).

Output field, returned in the response (IPN and Return URL).

Format string

Possible values

Value	Description
CREATED	The recurring payment has been successfully created. The recurring payment details are visible in the Merchant Back Office.
NOT_CREATED	The recurring payment has not been created and cannot be viewed in the Merchant Back Office.
ABANDONED	The request for creating a recurring payment has been abandoned by the buyer (debtor). The recurring payment has not been created and cannot be viewed in the Merchant Back Office.

Category Information about the subscription.

vads_redirect_error_message

Description	Allows to define the message that will appear before automatic redirection to the merchant website if the payment has been declined. *Input field.*
Format	ans..255
Error code	37
Category	Redirection to the merchant website.

vads_redirect_error_timeout

Description	Allows to define a delay in seconds before an automatic redirection to the merchant website at the end of a declined payment. The value of the field is between **0** and **300** s. After this delay, the buyer will be redirected to the URL populated in the **vads_url_refused** field. If the parameter is not populated, the buyer will be redirected to the return URL entered in the **vads_url_return** field or to the return URL entered in the Merchant Back Office. If the return URL is not set, the buyer will be redirected to the merchant website. *Input field.*
Format	n..3
Error code	36
Category	Redirection to the merchant website.

vads_redirect_success_message

Description	Allows to specify the message that will appear upon automatic redirection to the merchant website. *Input field.*
Format	ans..255
Error code	35
Category	Redirection to the merchant website.

vads_redirect_success_timeout

Description	Allows to define a delay in seconds before an automatic redirection to the merchant website at the end of an accepted payment. Its value is between 0 and 300s. After this delay, the buyer will be redirected to the URL populated in the **vads_url_success** field. If the parameter is not populated, the buyer will be redirected to the return URL entered in the **vads_url_return** field or to the return URL entered in the Merchant Back Office. If the return URL is not set, the buyer will be redirected to the merchant website. *Input field.*
Format	n..3
Error code	34
Category	Redirection to the merchant website.

vads_requestor

Description	Allows to modify the value of the "**Aceite**" field for a Boleto Bancario. The **Aceite** field can have two values: • **N** (= No) Default value The boleto has been generated without an official authorization of the buyer by means of a signed document. • **S** (= Yes) The buyer's authorization is mandatory as the signed document will serve as evidence of debt. *Input and output field, returned in the response (IPN and Return URL).*
Format	enum
Possible values	• BANK Means that the **S** (= Yes) value will be applied in the **Aceite** field. • MERCHANT Means that the **N** (= No) value will be applied in the **Aceite** field.

Category	Transaction details.

vads_result

Description

Return code of the requested action.

Output field, returned in the response (IPN and Return URL).

Format

n2

Possible values

Value	Description
00	Action successfully completed.
05	Action rejected.
17	Action cancelled by buyer.
30	Format error in merchant request. The transaction has been created. For more information, please see the vads_extra_result field .
96	Action refused due to technical error.

Category

Technical information.

vads_return_mode

Description

Allows to specify the data transmission method used while returning to the merchant website.

Input field.

Format

enum

Error code

48

Possible values

Field name	Value	Description
vads_return_mode	absent, empty or **NONE**	No parameters will be transmitted to the Return URL.
	GET	The return fields will be transmitted to the return URL in an HTTP GET form (in the "query string").
	POST	The return fields will be transmitted to the return URL in an HTTP POST form. If the return to the shop in done from an environment **other than https**, a security pop-up message will be displayed to the buyer.

Category

Redirection to the merchant website.

vads_risk_analysis_result

Description Returns the result of the risk management process performed by an external system (Konduto, ClearSale, Cybersource, NOTO, etc.).

Output field, returned in the response (IPN and Return URL).

Format ans

Possible values

Values common to all risk analyzers	
INVALID_CREDENCIAL	Configuration problem of the risk management contract.
COMUNICATION_PROBLEM	Impossible to connect to the risk analyzer.
DATA_PROCESSING_PROBLEM	Problem occurred when processing the data being transmitted or the response of the risk management system.
MISSING_MANDATORY_ORDER_INFO	Order details are missing.
MISSING_MANDATORY_SHIPPING_INFO	Shipping details are missing.
MISSING_MANDATORY_SHIPPING_ADDRESS_INFO	Shipping address details are missing.
MISSING_MANDATORY_BILLING_INFO	Billing details are missing.
MISSING_MANDATORY_BILLING_ADDRESS_INFO	Billing address details are missing.
MISSING_MANDATORY_CARD_INFO	Payment method details are missing.
MISSING_MANDATORY_CUSTOMER_INFO	Buyer details are missing.

Values returned by ClearSale	
APA	The transaction is automatically approved according to the defined parameters.
APM	The transaction is manually approved by an analyst.
RPM	The order is reproved due to missing information related to the buyer in conformity with the policy in force.
AMA	Waiting for manual analysis. The order is waiting to be analyzed.
ERR	Error
NVO	New order. Waiting to be processed and classified.
SUS	Order manually suspended. The order is suspended for suspected fraud.
CAN	Order is canceled. The order has been canceled by the buyer.
FRD	Fraud confirmed by the credit card operator or the cardholder.
RPA	Order automatically declined. The order has been automatically declined in accordance with the parameters of the external risk analyzer.
RPP	Order automatically declined. The order is reproved based on the customer or ClearSale policy.

Category Transaction details.

vads_risk_assessment_result

Description

Returns the list of actions performed on the transaction, following the activation of the advanced risk assessment activated in the Merchant Back Office.

When triggering multiple rules, the **vads_risk_assessment_result** field will consist of multiple keywords separated by a ";".

Example:

vads_risk_assessment_result="ENABLE_3DS;MANUAL_VALIDATION"

Output field, returned in the response (IPN and Return URL).

Format ans

Possible values

Values	Description
ENABLE_3DS	• The risk module has requested an authentication with cardholder interaction (challenge).
DISABLE_3DS	• The risk module has requested an authentication without cardholder interaction (frictionless).
NO_PREFERENCE	• The risk module has requested 3DS authentication. The choice of the preference is transferred to the card issuer.
NO_CHALLENGE_REQUESTED	• The risk module has requested an authentication without cardholder interaction (frictionless).
CHALLENGE_REQUESTED	• The risk module has requested an authentication with cardholder interaction (challenge).
CHALLENGE_MANDATE	• The risk module has requested an authentication with cardholder interaction (challenge for regulatory reasons) for regulatory reasons.
MANUAL_VALIDATION	• The transaction has been created via manual validation. • The payment capture is temporarily blocked to allow the merchant to perform all the desired verification processes.
REFUSE	• The transaction is refused.
RUN_RISK_ANALYSIS	• Call to an external risk analyzer if the Merchant has a contract. Refer to the FraudManagement.RiskAnalysis object in TransactionDetails field to identify the list of possible values and their description.
INFORM	• A warning message appears. • The Merchant is notified that a potential problem has been identified. • The Merchant is informed via one or several notification center rules (Instant Payment Notification (IPN), e-mail or SMS).

Category Transaction details

vads_risk_control

Description

Allows to define the result of the risk management process.

Output field, returned in the response (IPN and Return URL).

Format

control1=result1;control2=result2

Possible values

Value	Description
CARD_FRAUD	Verifies whether the cardholder's card number is on the card greylist.
SUSPECT_COUNTRY	Verifies whether the cardholder's card number is on the list of forbidden countries.
IP_FRAUD	Verifies whether the cardholder's IP address is on the IP greylist.
CREDIT_LIMIT	Verifies the purchase frequency and amounts for the same card number, or the maximum amount of an order.
BIN_FRAUD	Verifies whether the BIN code of the card is on the greylist for BIN codes.
ECB	Verifies whether the buyer's card is an "e-carte bleue".
COMMERCIAL_CARD	Verifies whether the buyer's card is a corporate credit card.
SYSTEMATIC_AUTO	Verifies whether the buyer's card is a MAESTRO or VISA ELECTRON credit card.
INCONSISTENT_COUNTRIES	Verifies whether the country of the IP address, the country of the payment card and the country of residence of the buyer match.
NON_WARRANTY_PAYMENT	Verifies the liability shift of the transaction.
SUSPECT_IP_COUNTRY	Verifies whether the cardholder's country, identified by his/her IP address, is on the list of forbidden countries.

The possible values for **result** are:

Value	Description
OK	OK
WARNING	Informational control failed
ERROR	Blocking control failed

Category

Transaction details

vads_sequence_number

Description

Transaction sequence number.

Case of single payment (vads_payment_config=SINGLE)

vads_sequence_number is populated with 1 in case of single payment.

However, if the merchant has authorized several payment attempts after a rejected payment, the sequence number will be incremented upon each new attempt.

*The number of additional attempts after a rejected payment can be configured via the Merchant Back Office (menu **Settings** > **Shop** > **Configuration**).*

If **vads_payment_config = SINGLE**:

vads_url_check_src	vads_sequence_numbe	Description
PAY	1	Payment made in 1 attempt

vads_url_check_src	vads_sequence_numbe	Description
	2	Payment made in 2 attempts
	3	Payment made in 3 attempts
BATCH_AUTO	1	Deferred payment made in 1 attempt
	2	Deferred payment made in 2 attempts
	3	Deferred payment made in 3 attempts

Case of installment payment (vads_payment_config=MULTI)

With the application of Soft Decline, the **vads_sequence_number** field no longer allows to easily identify the first installment of a payment in installments.

To identify the first payment of a series, view the description of the **vads_occurrence_type** field.

Case of a cascading payment (vads_card_brand=MULTI)

In case of a cascading payment (the cart items are paid with several payment methods), the **vads_sequence_number** field is always set to **0**.

The **vads_payment_seq** field describes each transaction in a table in JSON format.

The **transaction[x].sequence_number** attribute takes the same values as the **vads_sequence_number** field, as described for the case of single payment.

Note:

The **vads_sequence_number** field is not returned in the response when a payment is canceled or abandoned.

Output field, returned in the response (IPN and Return URL).

Category Transaction details.

vads_ship_to_city

Description Allows to specify the city for shipping.

Input and output field, returned in the response (IPN and Return URL).

Format an..128

Error code 83

Category Shipping details.

vads_ship_to_country

Description Allows you to specify the buyer's country code in the ISO 3166 standard.

Input and output field, returned in the response (IPN and Return URL).

Format a2

Error code 86

Examples of possible values

Code	Country	Code	Country
AT	Austria	**GP**	Guadeloupe
CI	Ivory Coast	**MQ**	Martinique
DE	Germany	**NC**	New Caledonia
ES	Spain	**PF**	French Polynesia
FR	France	**PM**	St. Pierre and Miquelon
FR	Corsica	**US**	United States of America

Category Shipping details.

vads_ship_to_delay

Description Allows to define the speed depending on the shipping method when **vads_ship_to_speed** is set to **PRIORITY**.

Input field.

Format enum

Error code 127

Possible values

- **INFERIOR_EQUALS** for a shipping delay inferior or equal to 1 hour.
- **SUPERIOR** for a shipping delay exceeding 1 hour.
- **IMMEDIATE** for an immediate shipping.
- **ALWAYS** for a 24/7 shipping delay.

Category Shipping details.

vads_ship_to_delivery_company_name

Description Allows to define the name of the transporter.

Input field.

Format ans..127

Error code 96

Category Shipping details.

vads_ship_to_district

Description Allows to define the shipping district.

Input and output field, returned in the response (IPN and Return URL).

Format ans..127

Error code	115
Category	Shipping details.

vads_ship_to_first_name

Description	Allows to specify the buyer's first name. *Input field.*
Format	ans..63
Error code	106
Category	Shipping details.

vads_ship_to_last_name

Description	Allows to specify the buyer's last name. *Input field.*
Format	ans..63
Error code	107
Category	Shipping details.

vads_ship_to_legal_name

Description	Company name of the shipping location. *Input and output field, returned in the response (IPN and Return URL).*
Format	an..100
Error code	125
Category	Shipping details.

vads_ship_to_name

Description	Allows to specify the buyer's last name. **Deprecated**. Please use **vads_ship_to_first_name** and **vads_ship_to_last_name** fields. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..63
Error code	80
Category	Shipping details.

vads_ship_to_phone_num

Description	Allows to specify buyer's phone number used for shipping. Accepts all formats:

Examples:

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

Input and output field, returned in the response (IPN and Return URL).

Format	ans..32
Error code	87
Category	Shipping details.

vads_ship_to_speed

Description	Allows to specify the shipping mode. *Input field.*
Format	enum
Error code	95
Possible values	• **STANDARD** (Specific to 3x 4x Oney) • **EXPRESS** (Specific to 3x 4x Oney) • **PRIORITY** (Specific to 3x 4x Oney) ***Note:*** *The use of **PRIORITY** as a value implies that the **vads_ship_to_delay** field will be used.*
Category	Shipping details.

vads_ship_to_status

Description	Allows to specify the type of the shipping address. *Input and output field, returned in the response (IPN and Return URL).*
Format	enum
Error code	93
Possible values	**PRIVATE**, **COMPANY**
Category	Shipping details.

vads_ship_to_state

Description	Allows to specify the buyer's state for shipping. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..127
Error code	84
Category	Shipping details.

vads_ship_to_street

Description	Allows to specify the buyer's address. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..255 ***Note:*** *The > and < special characters are not authorized.*
Error code	81
Category	Shipping details.

vads_ship_to_street2

Description	Allows to specify the second line of the buyer's address. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..255 ***Note:*** *The > and < special characters are not authorized.*
Error code	82
Category	Shipping details.

vads_ship_to_street_number

Description	Allows to specify the shipping street number. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..64
Error code	114
Category	Shipping details.

vads_ship_to_type

Description	Allows to specify the shipping type. *Input field.*
Format	enum
Error code	94
Possible values	• **RECLAIM_IN_SHOP** for picking up the item at the shop. • **RELAY_POINT** for using a third-party pick-up network (Kiala, Alveol, etc.). • **RECLAIM_IN_STATION** for picking up the item at an airport, a guard or a travel agency. • **PACKAGE_DELIVERY_COMPANY** for shipping by the transporter (Colissimo, UPS, etc.). • **ETICKET** for sending an electronic ticket, download.
Category	Shipping details.

vads_ship_to_user_info

Description	Information about the user who made the payment. This parameter will be returned with the response and will include the value transmitted in the request. ***Note:*** *For backward compatibility, it is possible to use this field to set the CPF/CNPJ (legal identifier in a numeric format between 11 and 20 digits long) required by the ClearSale risk management module. However,* ***vads_cust_national_id*** *field can be used.* The input and output field, returned in the response (IPN and Return URL).
Format	ans..255
Error code	116
Category	Shipping details.

vads_ship_to_zip

Description	Allows to specify the buyer's postal code. *Input and output field, returned in the response (IPN and Return URL).*
Format	an..64
Error code	85
Category	Shipping details.

vads_shipping_amount

Description	Allows to transmit the shipping fees for the whole order. *Input field.*
Format	n..12
Error code	109
Category	Shipping details.

vads_shop_name

Description	Allows to define the shop name as it appears in the summary at the end of payment, the receipt and the payment confirmation e-mails. *The input and output field, returned in the response (IPN and Return URL).*
Format	ans..127
Error code	72
Category	Customization of the payment page.

vads_shop_url

Description	URL that appears on the payment page and in payment confirmation e-mails. This setting overrides the default value of your shop. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..1024
Error code	73
Category	Customization of the payment page.

vads_site_id

Description	Mandatory parameter. Value generated during the subscription to the payment gateway. Its value can be seen in the interface of the Merchant Back Office via **Settings** > **Shop** > **Keys** by all authorized individuals. If the value is incorrect, the buyer will get an error message in their browser when making the payment. It becomes impossible to make the payment and the transaction is definitively lost. A warning e-mail is then sent to the shop administrator. It contains the form that the gateway was unable to process with the value of the signature. *Input and output field, returned in the response (IPN and Return URL).*
Format	n8
Error code	02
Category	Technical informations.

vads_subscription

Description	Optional parameter used for creating a recurring payment. It designates the ID of the recurring payment ID to create. There are two choices: • The payment gateway manages the IDs. In this case, this parameter must not be populated. In case the subscription is successfully created, the response will contain the value generated by the payment gateway. • The merchant website manages the IDs. In this case, this parameter must be populated with the desired value of the subscription ID. There is no uniqueness check on the subscription ID. When creating a subscription, the merchant site can fill **vads_subscription** with an already existing value.

It is possible to create multiple subscriptions, associated with the same token, with the same subscription ID.

Input and output field, returned in the response (IPN and Return URL).

Format Two possible formats:

- **an32**: when the identifier is generated by the gateway
- **ans..50**: when the identifier is generated by the merchant

Error code 63

Category Recurring payment details.

vads_sub_amount

Description

Mandatory parameter used for creating a recurring payment.

It refers to the amount of each installment except the ones that will be defined by **vads_sub_init_amount_number**.

The value cannot be negative, empty, or equal to 0.

The value must be expressed in the smallest currency unit (cent for euro).

Example: for a transaction of 10 euros and 28 cents, the value of the parameter is 1028.

Input field.

Format

n..12

Error code

65

Category

Information about the subscription.

vads_sub_currency

Description

Mandatory parameter used for creating a recurring payment.

It indicates the currency to use for the recurring payment, in compliance with the ISO 4217 standard.

Input and output field, returned in the response (IPN and Return URL).

Format

n3

Examples of possible values

The possible currencies are:

Currency	ISO 4217 encoding	Number of digits after the decimal point
Australian Dollar (AUD)	036	2
Cambodian Riel (KHR)	116	0
Canadian Dollar (CAD)	124	2
Chinese Yuan (Renminbi) (CNY)	156	1
Czech Crown (CZK)	203	2
Danish Crown (DKK)	208	2
Hong Kong Dollar (HKD)	344	2
Hungarian Forint (HUF)	348	2
Indian Rupee (INR)	356	2
Indonesian Rupiah (IDR)	360	2
Japanese Yen (JPY)	392	0
South Korean Won (KRW)	410	0
Kuwaiti Dinar (KWD)	414	3
Malaysian Ringgit (MYR)	458	2
Mexican Peso (MXN)	484	2
Moroccan Dirham (MAD)	504	2
New Zealand dollar (NZD)	554	2
Norwegian Crown (NOK)	578	2
Philippine Peso (PHP)	608	2
Russian Ruble (RUB)	643	2
Singapore Dollar (SGD)	702	2
South-African Rand (ZAR)	710	2
Swedish Crown (SEK)	752	2

Currency	ISO 4217 encoding	Number of digits after the decimal point
Swiss Franc (CHF)	756	2
Thai Baht (THB)	764	2
Tunisian Dinar (TND)	788	3
Pound Sterling (GBP)	826	2
US Dollar (USD)	840	2
Taiwan New Dollar (TWD)	901	2
Romanian Leu (RON)	946	2
New Turkish Lira (TRY)	949	2
Euro (EUR)	978	2
Polish Zloty (PLN)	985	2
Brazilian Real (BRL)	986	2

Error code 67

Category Information about the subscription.

vads_sub_desc

Description Mandatory parameter used for creating a recurring payment.

It designates the recurring payment rule to be applied.

The expected value for this parameter is a chain of characters that comply with the **iCalendar** (Internet Calendar) specification, described in RFC5545 (see *http://tools.ietf.org/html/rfc5545*).

Among other aspects, this specification allows to define complex recurring payment rules via the **RRULE** property.

For technical reasons, it is not possible to define recurring payment periods that are shorter than one day.

The keywords "SECONDLY" / "MINUTELY" / "HOURLY" are not taken into account.

Examples:

- To program installment payments taking place on the last day of each month for 12 months, the rule is:

  RRULE:FREQ=MONTHLY;BYMONTHDAY=28,29,30,31;BYSETPOS=-1;COUNT=12

  This rule means that if the current month does not have 31 days, the machine will take the 30th into account. If there is no 30th day in a month, the machine will take the 29th into account, and so on until the 28th.

  Another version of this rule: **RRULE:FREQ=MONTHLY;COUNT=5;BYMONTHDAY=-1**

- To program installment payments on the 10th of each month for 12 months, the rule is: **RRULE:FREQ=MONTHLY;COUNT=12;BYMONTHDAY=10**

- To program installment payments every three months up to December 31st, 2016.
  RRULE:FREQ=YEARLY;BYMONTHDAY=-1;BYMONTH=1,4,7,10;UNTIL=20161231

  The installment payments will be due on the first day of January, April, July and October each year. The total number of installments depends on the recurring payment start date (see **vads_sub_effect_date** parameter).

- In order to define a weekly recurring payment to be made every Monday: **RRULE:FREQ=WEEKLY;BYDAY=MO**

  The installments will be made every Monday. Note that the first installment will occur the nearest Monday.

- In order to define a weekly recurring payment: **RRULE:FREQ=WEEKLY**

  The installments will occur on the same day if the due date is set to “today”, then every 7 days.

- In order to define a recurring payment every two weeks on Monday, with maximum 4 installments: **RRULE:FREQ=WEEKLY;INTERVAL=2;COUNT=4;BYDAY=MO**

- In order to define a recurring payment every two weeks, on the same day and every 7 days: **RRULE:FREQ=WEEKLY;INTERVAL=2;**

- For more information and examples, visit *http://recurrance.sourceforge.net/*.

Input and output field, returned in the response (IPN and Return URL).

Format string

Error code 64

Category Recurring payment details.

vads_sub_effect_date

Description	Subscription start date (or effective date) in the UTC time zone, in **YYYYMMDD** format. Mandatory parameter used for creating a recurring payment. This parameter does not always match with the date of the first installment that depends only on the **vads_sub_desc** parameter. **Example:** for 1 February 2015, use 20150201. *Input and output field, returned in the response (IPN and Return URL).*
Format	n8
Error code	69
Category	Subscription details.

vads_sub_init_amount

Description	Optional parameter used for creating a subscription. Represents the amount of the first installments of the recurring payment. The number of these first installments is defined by the **vads_sub_init_amount_number** parameter. This amount is presented in the currency defined by the **vads_sub_currency** parameter and is expressed in its smallest unit (cent for euro). E.g.: for an amount of 10 euros and 28 cents, the value of the parameter is 1028. The value may be empty but cannot be negative or equal to 0. *Input and output field, returned in the response (IPN and Return URL).*
Format	n..12
Error code	66
Category	Subscription details

vads_sub_init_amount_number

Description	Optional parameter used for creating a recurring payment. Number of installments for which the **vads_sub_init_amount** should be applied. Once these installments will have expired, the **vads_sub_amount** field will be used. **Example**: to define a recurring payment with the first 3 installments of EUR 45.25 , and the rest of the installments of EUR 75.90, the following values will be used: • vads_sub_currency = 978 • vads_sub_init_amount_number = 3 • vads_sub_init_amount = 4525 • vads_sub_amount = 7590 *Input and output field, returned in the response (IPN and Return URL).*
Format	n..3

Code erreur	68
Category	Information about the subscription.

vads_submerchant_address

Description	Address of the sub-merchant. Transmitted by the payment facilitator. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..255
Error code	180
Category	Sub-merchant details.

vads_submerchant_address2

Description	Address line 2 of the sub-merchant. Transmitted by the payment facilitator. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..255
Error code	181
Category	Sub-merchant details.

vads_submerchant_city

Description	City of the sub-merchant. Transmitted by the payment facilitator. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..128
Error code	183
Category	Sub-merchant details.

vads_submerchant_company_type

Description	Company type of the sub-merchant. Transmitted by the payment facilitator. This field is used to specify which type the Legal Number corresponds to. Different rules may apply depending on the purchaser. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..255
Error code	188
Category	Sub-merchant details.

vads_submerchant_country

Description	Country of the sub-merchant's address (ISO 3166 alpha-2 standard). Transmitted by the payment facilitator.

Input and output field, returned in the response (IPN and Return URL).

Format	a2
Error code	184
Category	Sub-merchant details.

vads_submerchant_facilitatorId

Description	Payment Facilitator ID. Transmitted by the payment facilitator. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..128
Error code	192
Category	Sub-merchant details.

vads_submerchant_legal_number

Description	Legal Entity Identifier of the sub-merchant according to the field vads_submerchant_company_type . Transmitted by the payment facilitator. The identifier depends on the country of the sub-merchant. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..24
Error code	189
Category	Sub-merchant details.

vads_submerchant_mcc

Description	Merchant Category Code of the sub-merchant. Transmitted by the payment facilitator. Allows to identify the activity of the sub-merchant. *Input and output field, returned in the response (IPN and Return URL).*
Format	n4
Error code	185
Category	Sub-merchant details.

vads_submerchant_mid

Description	Acquirer contract number (MID) of the sub-merchant. Transmitted by the payment facilitator. *Input and output field, returned in the response (IPN and Return URL).*
Format	n..50
Error code	186

Category	Sub-merchant details.

vads_submerchant_name

Description	Legal name of the sub-merchant. Transmitted by the payment facilitator. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..255
Error code	177
Category	Sub-merchant details.

vads_submerchant_phone

Description	Phone number of the sub-merchant. Transmitted by the payment facilitator. Accepts all formats: Examples: • 0123456789 • +33123456789 • 0033123456789 • (00.571) 638.14.00 • 40 41 42 42 *Input and output field, returned in the response (IPN and Return URL).*
Format	an..32
Error code	179
Category	Sub-merchant details.

vads_submerchant_soft_descriptor

Description	(soft descriptor) of the sub-merchant that appears on the buyer's bank statement. Transmitted by the payment facilitator. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..255
Error code	187
Category	Sub-merchant details.

vads_submerchant_state

Description	Region of the sub-merchant address. Transmitted by the payment facilitator. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..128
Error code	191

Category	Sub-merchant details.

vads_submerchant_url

Description	URL of the sub-merchant. Transmitted by the payment facilitator. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..1024
Error code	178
Category	Sub-merchant details.

vads_submerchant_zip

Description	Zip code of the sub-merchant. Transmitted by the payment facilitator. *Input and output field, returned in the response (IPN and Return URL).*
Format	an..64
Error code	182
Category	Sub-merchant details.

vads_tax_amount

Description	Parameter that allows to define the amount of taxes for the entire order. The value must be specified in the smallest currency unit (cents for euro). *Input and output field, returned in the response (IPN and Return URL).*
Format	n..12
Error code	108
Category	Order details.

vads_tax_rate

Description	Allows to define the tax rate (VAT) applied to the order. The value must be expressed in XX.XX format, with a **dot** as the separator and without the **%** suffix. Example: "19.00" for 19%. *Input and output field, returned in the response (IPN and Return URL).*
Format	XX.XX
Error code	153
Category	Order details.

vads_tax_refund_amount

Description

This field is used in Uruguay and it corresponds to the tax credit amount allocated to the merchant for the transaction.

The value is specified in the smallest currency unit (cents for euro).

The field is present only if the acquirer returns this information.

Output field, returned in the response (IPN and Return URL).

Format

n..12

Category

Transaction details.

vads_theme_config

Description

Allows to personalize certain elements on the payment page, such as the custom template to be used, the button labels and some messages.

This parameter provides a list of keywords (codes), each associated with a value, that correspond to elements on payment pages.

Example:

```
vads_theme_config="SUBMIT_BUTTON_LABEL=PAY;TICKET_LABEL=PAYMENT RECEIPT"
```

See *Back Office user manual - Advanced customizationfor more details on payment page personalization.*

Input field.

Format

map

Error code

32

Possible values

Code	Description
Features	
RESPONSIVE_MODEL	Allows to override the custom template to be applied to the payment pages. Example of use: `vads_theme_config="RESPONSIVE_MODEL=Model_1"` The use of custom templates requires the activation of the “Advanced customization” option.
RESPONSIVE_MAIL_MODEL	Allows to override the custom template to be used for e-mails. Example of use: `vads_theme_config="RESPONSIVE_MAIL_MODEL=Model_1"` The use of custom templates requires the activation of the “Advanced customization” option.
HIGH_CONTRAST_MODE	Allows to enable the high contrast mode to enhance color contrast and display the payment page in black and white. Possible values: "**true**" or "**false**". Example of use: `vads_theme_config="HIGH_CONTRAST_MODE=true"`

Code	Description
SIMPLIFIED_DISPLAY	Allows to reduce the volume of data to be loaded during the display of the payment page. Deletes the language and logo selector from the footer. Recommended for iframe and In-app integrations. Possible values: "**true**" or "**false**". Example of use: `vads_theme_config="SIMPLIFIED_DISPLAY=true"`
FORM_TARGET	Allows to define or display the return page at the end of payment. Possible values: • _blank: in a new window or a new tab • _self: in the current frame • _parent: in the parent frame • _top: on the whole page • *framename*: in a specified frame Example of use: `vads_theme_config="FORM_TARGET=_top"`
3DS_LOGOS	Allows to mask the “Verified By Visa” and “Mastercard Secure Code” logos on the card detail entry page. Possible values: "**true**" or "**false**". Example of use: `vads_theme_config="3DS_LOGOS=false"`
Button labels	
SUBMIT_BUTTON_LABEL	Allows to edit the label of the “VALIDATE” button. Example of use: `vads_theme_config="SUBMIT_BUTTON_LABEL=PAY"`
CANCEL_FOOTER_MSG_RETURN	The label of the “Cancel and return to the shop” button on the page of payment method selection, the card detail entry page, and the result page in case of payment failure. Example of use: `vads_theme_config="CANCEL_FOOTER_MSG_RETURN=CANCEL"`
SUCCESS_FOOTER_MSG_RETURN	The label of the “Return to the shop” button on the result page in case of successful payment. Example of use: `vads_theme_config="SUCCESS_FOOTER_MSG_RETURN=RETURN"`
TICKET_LABEL	The label of the “RECEIPT” button on the result page in case of successful payment. Example of use: `vads_theme_config="TICKET_LABEL=PAYMENT RECEIPT"`
Messages	
MERCHANT_ MESSAGE	Allows to display a message above the transaction summary. Requires for the **Display a custom message** checkbox to be checked via **Settings** > **Customization** > **Payment pages** tab > **Logo** group. Example of use: `vads_theme_config="MERCHANT_MESSAGE=Transaction summary"`

Code	Description
SECURE_ MESSAGE	Default value: *The address of this payment website prefixed with https indicates that you are on a secure page and can safely proceed to your payment*. Example of use: `vads_theme_config="SECURE_ MESSAGE=You are on a website secured by TLS1.2. You can safely proceed to payment."`
SECURE_MESSAGE_REGISTER	Default value: *The address of this payment website prefixed with https indicates that you are on a secure page and can safely enter your bank details*.
REGISTER_ON_PAYMENT	Allows to customize the text of the checkbox that appears during ASK_REGISTER_PAY. Default value: *I would like to register my payment method details for a future purchase*
Labels that appear on the receipt and the payment pages	
SITE_ID_LABEL	Default value: *Merchant ID*
ORDER_ID_LABEL	Default value: *Order reference*
TRANSACTION_ID_LABEL	Default value: *Transaction number*
TRANSACTION_AMOUNT_LABEL	Default value: *Amount*
MULTI_DATE_LABEL	Default value: *Sale date* Information displayed only during an installment payment.
CUST_ID_LABEL	Default value: *Buyer reference* Information displayed only during a payment by token.
CUST_ADRESS_NUMBER_LABEL	Default value: *Street number* Information displayed only during a payment by token.
CUST_ADRESS_LABEL	Default value: *Address* Information displayed only during a payment by token.
CUST_ADRESS2_LABEL	Default value: *Address line 2* Information displayed only during a payment by token.
CUST_DISTRICT_LABEL	Default value: *District* Information displayed only during a payment by token.
CUST_CITY_LABEL	Default value: *City* Information displayed only during a payment by token.
CUST_COUNTRY_LABEL	Default value: *Country* Information displayed only during a payment by token.
CUST_PHONE_LABEL	Default value: *Phone* Information displayed only during a payment by token.
CUST_NAME_LABEL	Default value: *Buyer's last name* Information displayed only during a payment by token.
RECURRENCE_AMOUNT_LABEL	Default value: *Amount per installment* Information displayed only during a payment by token.
RECURRENCE_INIT_AMOUNT_ NUMBER_LABEL	Default value: *Number of installments of initial amount* Information displayed only during a payment by token.
RECURRENCE_INIT_AMOUNT_LABEL	Default value: *Initial amount of the recurring payment* Information displayed only during a payment by token.
SHOP_LABEL	Default value: *SHOP* Information displayed only on the PDF receipt.
SITE_URL_LABEL	Default value: *URL address* Information displayed only on the PDF receipt.
CUST_LANGUAGE	Default value: *Language* Information displayed only on the PDF receipt.

Category Payment page customization

vads_threeds_auth_type

Description

Indicates the authentication type of the cardholder.

Output field, returned in the response (IPN and Return URL).

Format

enum

Possible values

- "Empty" if the buyer is not correctly authenticated.
- **FRICTIONLESS**: cardholder authentication without interaction with the ACS. Value returned only in 3DS v2.
- **CHALLENGE**: interactive cardholder authentication (entering an OTP or replying to a series of questions). Value returned in 3DS v2.
- **DATA ONLY**: authentication handled by the DS without client interaction. This is an option of the EMV 3D Secure protocol, available only with Mastercard.

Category

Authentication of cardholder.

vads_threeds_cavv

Description

Designates the cardholder's authentication through the ACS. Its value is populated by 3DS authentication server (ACS) when the buyer has been correctly authenticated (vads_threeds_status equals "Y" or "A").

Output field, returned in the response (IPN and Return URL).

Format

ans..28

Category

Authentication of cardholder.

vads_threeds_cavvAlgorithm

Description

Algorithm used by the ACS to generate the CAVV value.

Its value is populated by 3DS authentication server (ACS) when the buyer has been correctly authenticated (vads_threeds_status equals "Y" or "A").

Output field, returned in the response (IPN and Return URL).

Format

an1

Possible values

Value	Description
0	HMAC
1	CVV
2	CVV_ATN
3	MasterCard SPA
A	AV-CB

Category

Authentication of cardholder.

vads_threeds_eci

Description

Indicates the E-Commerce index.

It is populated by the 3DS authentication server (ACS) when the buyer has been correctly authenticated (vads_threeds_status equals « Y » or « A »).

DS	status = Y	status = A	status = U	status = N
VISA,CB, ELO, AMEX, DINERS, DISCOVER	5	6	7	-
MasterCard	02	01	-	-

In case of authentication without payment (e.g. card registration), Mastercard can return the following values:

DS	status = Y	status = A	status = U	status = N
MasterCard	N2	-	N0	N0

Output field, returned in the response (IPN and Return URL).

Format an..2

Category Authentication of cardholder.

vads_threeds_enrolled

Description Indicates the enrollment status of the cardholder. Its value is populated by the VISA and MASTERCARD (DS) servers during 3D Secure authentication.

Output field, returned in the response (IPN and Return URL).

Format a1

Possible values

Value	Description
Y	Cardholder enrolled, 3DS authentication possible. *Note: In the Merchant Back Office, the ENROLLED value is displayed among transaction details (Authentication tab).*
N	Cardholder not enrolled. *Note: In the Merchant Back Office, the NOT_ENROLLED value is displayed among transaction details (Authentication tab).*
U	Unable to verify the cardholder's enrollment status. *Note: In the Merchant Back Office, the UNAVAILABLE value is displayed among transaction details (Authentication tab).*

Category Authentication of cardholder.

vads_threeds_error_code

Description Final status of 3D Secure authentication.

This field is deprecated. It is replaced by the **vads_threeds_exit_status** field.

Output field, returned in the response (IPN and Return URL).

Format n..2

Category Authentication of cardholder.

vads_threeds_exit_status

Description Final status of the cardholder authentication process.

Populated by the payment gateway.

Output field, returned in the response (IPN and Return URL).

Format n..2

Possible values

Value	Description
0	Initial status
1	Status non-applicable (global, reason not detailed)
2	Status non-applicable (integrator disabled)
3	Not an e-commerce payment
4	Payment without 3DS
5	Merchant not enrolled, 3DS unavailable
6	A technical error has occurred during 3DS authentication, 3DS unavailable
7	**Cardholder not enrolled, 3DS unavailable**
8	Invalid signature
9	Problem caused by the ACS
10	**The 3DS authentication has been successfully completed**
11	The 3DS authentication has been completed via the integrator
12	Problem caused by DS
13	Timeout when connecting to DS
15	3DS disabled
16	Payment channel not available
98	Initialization of 3DS authentication OK
99	Unknown status

* These statuses concern 3DS payments without card detail entry (payment by token).

Category Authentication of cardholder.

vads_threeds_mpi

Description

This field indicates the merchant's desire to challenge the buyer with a strong authentication during a payment. The final decision to perform strong authentication is made by the issuer.

Input field.

Format

n1

Error code

50

Possible values

Use case	Values	Description
CHALLENGE: with cardholder interaction	**1**	**Deprecated**.
	3	**3DS Requestor Preference**: Allows to request strong authentication for the transaction.
	4	**Challenge request mandate**: Allows to indicate that, due to regulatory reasons, strong authentication is required for the transaction.
FRICTIONLESS: without cardholder interaction	**2***	Allows to Request an exemption from strong authentication: • Low value transactions. • Transactional Risk Analysis (TRA Acquéreur). • LRM (Low Risk Merchant). More informations: Table of exemptions, below.
No merchant preference	**0 or absent or empty**	The choice of the preference is transferred to the card issuer. If the issuer decides to perform an authentication without interaction (frictionless), the payment will be guaranteed.
	5	

***Table of exemptions** (value number 2):

Exemptions	Description
Low value transactions	In Europe, you can request an exemption from strong authentication, for transactions of less than €30, and within the limit of either 5 successive operations or a cumulative amount of less than €100. If the amount is higher than €30, the value transmitted by the merchant is ignored and the choice of the preference is transferred to the card issuer (No Preference). For payments made in a currency other than euro, a request for frictionless is transmitted to the issuer. If the frictionless request is accepted, the transaction does not benefit from liability shift dispute by the cardholder.. If the store does not have the "Frictionless 3DS2" option, the choice of the preference is transferred to the card issuer (No Preference).
Transactional Risk Analysis (TRA Acquéreur)	If your store has the "TRA Acquirer 3DS2" option, you can ask the issuer for an exemption from strong authentication if the amount is below the threshold set by your financial institution. If the frictionless request is accepted, the transaction does not benefit from liability shift dispute by the cardholder.. The "Acquirer 3DS2 TRA" activation option is subject to the prior agreement of your financial institution.
LRM (Low Risk Merchant)	CB offers the LRM (=Low Risk Merchant) program. This program is designed to meet the needs of very low-risk, high-

Exemptions	Description

	volume merchants. You can request an exemption from strong authentication:
	• If the amount is less than €100, the exemption is systematic for eligible merchants.
	• If the amount is between €100 and €250, an experiment is underway. To qualify, the merchant must :
	  • Have a CB contract.
	  • Be eligible for TRA acquéreur.
	  • Transmit the required values in the 3D Secure flow, according to the rules defined by the platform.
	If the frictionless request is accepted, the transaction does not benefit from liability shift dispute by the cardholder..
	To benefit from CB's LRM program, you must contact your customer advisorSociété Générale to obtain explicit approval.

Category Authentication of cardholder.

vads_threeds_sign_valid

Description Indicates the signature validity of the message containing the cardholder authentication result. Populated by the payment gateway.

Output field, returned in the response (IPN and Return URL).

Format n1

Possible values

Value	Description
empty	3DS unavailable.
0	Incorrect signature.
1	Correct signature.

Category Authentication of cardholder.

vads_threeds_status

Description

Defines the cardholder's authentication status. Populated by the 3DS authentication server (ACS) during the 3D Secure authentication.

Output field, returned in the response (IPN and Return URL).

Format

a1

Possible values

Value	Description
Y	Successful authentication. *Note: In the Merchant Back Office, the SUCCESS value is displayed (3D Secure tab in Transaction details).*
N	Authentication error. *Note: In the Merchant Back Office, the FAILED value is displayed (3D Secure tab in Transaction details).*
U	Authentication impossible. *Note: In the Merchant Back Office, the UNAVAILABLE value is displayed (3D Secure tab in Transaction details).*
A	Authentication attempt. *Note: In the Merchant Back Office, the ATTEMPT value is displayed (3D Secure tab in Transaction details).*

Category

Authentication of cardholder.

vads_threeds_xid

Description

Indicates the unique 3DS authentication reference.

It is populated by the authentication server (ACS) during the 3D Secure authentication process.

Output field, returned in the response (IPN and Return URL).

Format

ans..28

Category

Authentication of cardholder.

vads_tid

Description

Terminal ID. POS identifier defined within the acceptance contract.

Corresponds to the rank number (or logical number) for a CB contract.

Output field, returned in the response (IPN and Return URL).

Format

an..255

Category

Transaction details.

vads_token_id

Description

Payment order ID associated with the transaction.

Corresponds to the **paymentOrderId** fields in the REST API. It allows the Merchant to follow-up orders that they generated using the **PaymentOrder/Get** Web Service.

Output field, returned in the response (IPN and Return URL).

Format ans..255

Category Order details.

vads_totalamount_vat

Description Allows to define the total amount of taxes applied to the whole order.

The value must be specified in the smallest currency unit (cents for euro).

Input and output field, returned in the response (IPN and Return URL).

Format n..12

Error code 154

Category Order details.

vads_trans_date

Description

Mandatory parameter.

Corresponds to the timestamp in the YYYYMMDDHHMMSS format.

The timestamp must necessarily correspond to the current date and time, in the GMT + 0 (or UTC) time zone in 24h format.

Note:

If you are using REST payment web services, the equivalent of the vads_trans_date parameter is **transactions[0].transactionDetails.cardDetails.legacyTransDate**.

Input and output field, returned in the response (IPN and Return URL).

Format

n14

Error code

04

Frequent errors:

- The date has not been submitted in the YYYYMMDDHHMMSS format (year, month, day, hour, minute, second).
- The date is not based on the UTC time zone (Coordinated Universal Time).

  Make sure you use date functions in your programming language that will generate a UTC hour (e.g.: gmdate in PHP).
- The time must be calculated using the 24h format, not 12h.
- The buyer has waited for too long before clicking on **Pay**.
- The buyer was using browser history.

Category

Transaction details.

vads_trans_id

Description

Mandatory parameter.

It consists of 6 alphanumeric characters and must be unique for each transaction for a given shop on the same day.

***Note**: the uniqueness of the transaction identifier is based on the universal time (UTC).*

The merchant website must guarantee this uniqueness during same the day.

The range between 900000 and 999999 is reserved to the payment gateway, for the transactions made via:

- the Merchant Back Office (refunds, duplications, manual payment, etc.),
- the data collection form,
- a payment order.

Input and output field, returned in the response (IPN and Return URL).

Format

an6

Error code

03

Frequent errors:

The form is rejected:

- If the transmitted value contains less than 6 characters.
- If the value is null.
- If the field is absent.
- If an identical transaction number has already been sent on the same day.

  If the buyer clicks “Cancel and return to the shop”, the transaction number must be different for the next attempt as the previous one will be considered as already used.

  Otherwise, the “The transaction has been canceled” message will appear.

Category

Transaction details.

vads_trans_status

Description Allows to set the status of the transaction.

Output field, returned in the response (IPN and Return URL).

Format enum

Possible values

Value	Description
ABANDONED	**Abandoned** Payment abandoned by the buyer The transaction has not been created, and **therefore cannot be viewed in the Merchant Back Office**.
ACCEPTED	**Accepted**. Status of a VERIFICATION type transaction for which the authorization request or information request has been successfully completed. This status cannot evolve. Transactions with the **Accepted** status will never be captured.
AUTHORISED	**Waiting for capture** The transaction has been accepted and will be automatically captured at the bank on the expected date.
AUTHORISED_TO_VALIDATE	**To be validated** The transaction, created with manual validation, is authorized. The merchant must manually validate the transaction in order for it to be captured. The transaction can be validated as long as the expiration date of the authorization request has not passed. If the authorization validity period has been passed, the payment takes **Expired** status. This status is final.
CANCELLED	**Cancelled** The transaction has been canceled by the Merchant.
CAPTURED	**Captured** The transaction has been captured by the bank.
CAPTURE_FAILED	**Capture failed** Contact the technical support.
EXPIRED	**Expired** This status appears in the lifecycle of a payment with deferred capture. 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.
REFUSED	**Refused** The transaction is refused.
SUSPENDED	**Suspended** The capture of the transaction is temporarily blocked by the acquirer (AMEX GLOBAL or SECURE TRADING). Once the transaction has been correctly captured, its status changes to **CAPTURED**.
UNDER_VERIFICATION	**Control in progress** Waiting for the response from the acquirer. This status is temporary. A notification will be sent to the merchant website to inform the Merchant of the status change. Requires the activation of the Instant Payment Notification URL on batch change notification rule.
WAITING_AUTHORISATION	**Waiting for authorization** The capture delay in the bank exceeds the authorization validity period.

Value	Description
WAITING_AUTHORISATION_TO_VALIDATE	**To be validated and authorized** The capture delay in the bank exceeds the authorization validity period. A EUR 1 (or information request about the CB network if the acquirer supports it) authorization has been accepted. The merchant must manually validate the transaction for the authorization request and the capture to occur.

Category Transaction details.

vads_trans_uuid

Description Unique transaction reference generated by the payment gateway when creating a payment transaction.

Guarantees that each transaction is unique.

Output field, returned in the response (IPN and Return URL).

Format ans32

Example c3f8b11c2d464d7cae76057fa63e63eb

Category Transaction details.

vads_url_cancel

Description URL where the buyer will be redirected after having clicked on **Cancel and return to shop** before proceeding to payment.

Input field.

Format ans..1024

Error code 27

Category Redirection to the merchant website.

vads_url_check

Description URL of the page to notify at the end of payment. Overrides the value entered in the notification rule settings.

Note

This field should be used only in exceptional cases since:

- *This URL will only be used when calling the IPN URL,*
- *The override value will not be used if an automatic retry takes place.*

  It is not compatible with the execution of the request sent to the IPN from the Merchant Back Office. The called URL is the URL that has been set up in the notification rule (see chapter **Setting up notifications**).

  Input field.

Format ans..1024

Error code 33

Category	Redirection to the merchant website.

vads_url_check_src

Description

This parameter defines the source of the notification (also called IPN).

Output field, returned in the response (IPN and Return URL).

Format

enum

Possible values

Value	Description
PAY	Payment creation by form.
BO	Execution of the notification URL from the Merchant Back Office.
BATCH_AUTO	Authorization request for a payment that was awaiting authorization.
BATCH	Update of the transaction status after its synchronization on the acquirer side (case of notification on batch change).
REC	Payment resulting from a recurring payment.
MERCH_BO	Operation made via the Merchant Back Office.
RETRY	Automatic retry of the IPN.

Category

Redirection to the merchant website.

vads_url_error

Description

URL where the buyer will be redirected in case of an internal processing error.

Input field.

Format

ans..1024

Error code

29

Category

Redirection to the merchant website.

vads_url_post_wallet

Description

This field allows the merchant to transmit the URL to which the buyer will be redirected during a payment via a wallet in two steps.

This URL is used for transmitting information relative to the buyer's choice (e-mail, shipping address, payment method, etc.).

Based on these elements, the merchant can decide what to do (adjust the shipping fees, register the payment method, etc.) before allowing the buyer to finalize his or her payment.

The details will be transmitted to the merchant website via an html POST form.

Example: vads_url_post_wallet = https://mydomain-name.com/return_url

Note

*If the URL is inaccessible, the transaction cannot be finalized. After the payment session expires, a **rejected** transaction will be created. If the merchant has configured the notification rule for abandoned/canceled transactions, the merchant website will be notified about the reason of rejection via the **vads_payment_error** field. This field will be set to **149** indicating that the payment session has expired.*

*It will then become visible in the Merchant Back Office, in the **Event log** tab.*

	Input and output field, returned in the response (IPN and Return URL).
Format	ans..1024
Error code	138
Category	Redirection to the merchant website.

vads_url_referral

Description	**Deprecated field**. Use the **vads_url_refused** field. URL where the buyer will be redirected in case of a declined authorization (code 02 **Contact the card issuer**) after having clicked on **Return to shop**. *Input field.*
Format	ans..127
Error code	26
Category	Redirection to the merchant website.

vads_url_refused

Description	URL where the buyer will be redirected in case of a declined payment after having clicked on **Return to shop**. *Input field.*
Format	ans..1024
Error code	25
Category	Redirection to the merchant website.

vads_url_return

Description	Default URL to where the buyer will be redirected after having clicked on **Return to shop**, if **vads_url_error**, **vads_url_refused**, **vads_url_success** or **vads_url_cancel** is not set. If this field has not been transmitted, the Merchant Back Office configuration will be taken into account. It is possible to set up return URLs in TEST and PRODUCTION modes. These fields are called **Return URL of the shop in test mode** and **Return URL of the shop in production mode**; they can be viewed in **Settings** > **Shop** > **Configuration**. If no URL has been specified in the Merchant Back Office and in the form, the **Return to shop** button will redirect the buyer to the merchant website URL (**URL** field in the shop configuration section). *Input field.*
Format	ans..1024
Error code	28
Category	Redirection to the merchant website.

vads_url_success

Description	URL where the buyer will be redirected in case of an accepted payment after having clicked on **Return to shop**. *Input field.*
Format	ans..1024
Error code	24
Category	Redirection to the merchant website.

vads_use_case

Description	Allows you to specify that this is a payment upon shipment. This field is optional. To make a payment upon shipment, you must transmit one of the following use cases to the payment platform via this field: • SHIPMENT_MULTIPLE_AUTHORISATION Expected value for payment upon shipment with multiple authorization (payment on delivery). • SHIPMENT_SINGLE_AUTHORISATION Expected value for payment upon shipment single authorization (payment on order). Payment on dispatch is only compatible with CB, Mastercard and Visa. *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..50
Error code	N/A

Category	Transaction details.

vads_user_info

Description	Information about the user who made the payment. In the case of a form payment, this parameter will be resent with the response and will include the value transmitted in the request. In the case of a MOTO payment from the Merchant Back Office, this field will be valued with the user account (login) who made the payment. In the case of a payment order, this field will be populated with the user account (login) that created the order. ***Note:*** *For backward compatibility, it is possible to use this field to set the CPF/CNPJ (legal identifier in numeric format between 11 and 20 digits long) required by the ClearSale risk management module. However,* ***vads_cust_national_id*** *field can be used.* *Input and output field, returned in the response (IPN and Return URL).*
Format	ans..255
Error code	61
Category	Buyer details.

vads_validation_mode

Description

Specifies the transaction validation mode.

For more information on the transaction validation mode, see the chapter *Configuring the default validation mode* of the *sitemap.html* user manual.

Input and output field, returned in the response (IPN and Return URL).

Format

enum

Error code

05

Possible values

Value	Description
Missing or empty	Default configuration of the selected store (can be configured in the Merchant Back Office).
0	Transaction is automatically validated by the payment gateway.
1	The transaction must be validated manually by the merchant via their Merchant Back Office (or automatically via the **Transaction/Validate** Web Service function).

Category

Transaction details.

vads_version

Description

Mandatory parameter.

Version of the exchange protocol with the payment gateway.

Input and output field, returned in the response (IPN and Return URL).

Format

enum

Error code

01

Possible value

V2

Category

Technical details

vads_wallet

Description

This field allows the merchant to identify the type of wallet that was used for the payment.

Present only when a wallet was used for the payment.

List of existing Wallets

Value of the vads_wallet field	Wallet type
MASTERPASS	Masterpass by Mastercard

Output field, returned in the response (IPN and Return URL).

Format

an..127

Category

Payment method details.

vads_warranty_result

Description

Liability shift in case of accepted payment.

Output field, returned in the response (IPN and Return URL).

Format enum

Possible values

Value	Description
YES	The payment is guaranteed.
NO	The payment is not guaranteed.
UNKNOWN	Due to a technical error, the payment cannot be guaranteed.
Not specified	Liability shift not applicable.

Category Transaction details.

