



Accepting Titre-Restaurant cards

Implementation Guide

Document version 1.7.1

Contents

1. HISTORY OF THE DOCUMENT.....	4
2. TITRE-RESTAURANT CARDS.....	5
2.1. First generation cards.....	5
2.2. Second generation cards (CONECS scheme).....	7
2.3. Mixed cards.....	8
2.4. Summary.....	8
3. TECHNICAL INFORMATION.....	9
4. PREREQUISITES.....	10
5. PARTIAL CAPTURE AND VALIDATION MODE FOR DRIVE-THROUGH STORES.....	11
6. PAYMENT BY TOKEN.....	12
7. CASCADING PAYMENT.....	13
8. USING THE FORM WITH REDIRECT.....	14
8.1. Offering payment with a Titre-Restaurant card.....	14
8.2. Integration in the customer journey.....	18
8.3. Understanding the process of a Titre-Restaurant payment.....	19
8.3.1. Flow chart.....	19
8.3.2. The process of a single payment.....	21
8.3.3. The process of a Titre-Restaurant payment supplemented by another payment method.....	22
8.3.4. The process of payment with optional recording (ASK_REGISTER_PAY).....	23
8.3.5. The process of payment with mandatory recording (REGISTER_PAY).....	25
8.3.6. The process of a split payment with a registered card.....	27
8.4. Generating a payment form.....	28
8.4.1. Transmitting the amount payable by Titre-Restaurant card.....	30
8.4.2. Creating a payment request.....	31
8.4.3. Creating a payment request with registration of an electronic meal voucher.....	33
8.4.4. Creating a payment by token request.....	35
8.4.5. Creating a payment request with an update of an electronic meal voucher.....	36
8.4.6. Transmitting buyer details.....	37
8.4.7. Transmitting shipping details.....	38
8.4.8. Transmitting order details.....	39
8.5. Computing the signature.....	41
8.6. Sending the payment request.....	43
8.6.1. Redirecting the buyer to the payment page.....	43
8.6.2. Processing errors.....	44
8.7. Analyzing the payment result.....	46
8.7.1. Processing the response data.....	46
8.7.2. Analyzing the result of a cascading payment.....	48
8.7.3. Analyzing the result of the authorization request.....	51
9. USING THE JAVASCRIPT CLIENT (EMBEDDED FORM).....	52
9.1. Operating principle.....	52
9.2. Initializing a single payment request.....	54
9.3. Registering a meal voucher.....	56
9.4. Using a registered payment method.....	57
9.5. Using the buyer wallet.....	58
9.6. Analyzing the response.....	60
10. MANAGING CONECS TRANSACTIONS FROM THE MERCHANT BACK OFFICE.....	62

10.1. Viewing transaction details.....	62
10.2. Canceling a transaction.....	65
10.3. Modifying a transaction.....	66
10.4. Validating a transaction.....	67
11. MANAGING TRANSACTIONS USING REST WEB SERVICES.....	68
11.1. Retrieving transaction details.....	68
11.2. Canceling a transaction.....	69
11.3. Modifying a transaction.....	70
11.4. Validating a transaction.....	71
11.5. Zero-click payment (transaction initiated by the merchant).....	72
12. OBTAINING HELP.....	74

1. HISTORY OF THE DOCUMENT

Version	Author	Date	Comment
1.7.1	BPCE Payment Services	1/19/2023	<ul style="list-style-type: none"> Update of the chapter <i>Offering payment by Titre-Restaurant card</i>.
1.7	BPCE Payment Services	12/19/2022	<ul style="list-style-type: none"> Update of the payment limit Update of the Bimpli (ex Apetiz) logo
1.6	BPCE Payment Services	9/14/2021	<ul style="list-style-type: none"> Update of the chapter <i>Technical information</i>. Update of chapters on the payment process following the addition of payment without CVV. Clarification regarding the proposed tokens added in the chapter <i>The process of a split payment with a registered card</i>. Addition of the chapter <i>Partial capture and validation mode for drives</i>. Addition of integration via the JavaScript client. Addition of transaction management via REST Web Services.
1.5	BPCE Payment Services	9/14/2020	<ul style="list-style-type: none"> Addition of the <i>Technical information</i> chapter. Update of the capture delay. Clarifications in the chapter <i>Offering payment by Titre-Restaurant card</i>. Addition of the chapter <i>Integration in the customer journey</i>. Update of the vads_payment_cards field description in the <i>Creating a payment request</i> chapter. Update of the form example in the <i>Redirecting the buyer to the payment page</i> chapter.
1.4	BPCE Payment Services	2/11/2019	Document overhaul
1.3	BPCE Payment Services	1/14/2019	Initial version

This document and its contents are confidential. It is not legally binding. Any reproduction and / or distribution of all or part of this document or its content to a third party is strictly prohibited or subject to prior written authorization from BPCE Payment Services. All rights reserved.

2. TITRE-RESTAURANT CARDS

Titre-Restaurant cards work in the same way as prepaid cards. They provide access to an account where employers regularly recharge the electronic meal vouchers of their employees, who will be able to pay for their purchases in all the affiliated establishments (restaurants, food shops, caterers, supermarkets, bakeries, butchers...), .

The minimum amount that can be paid using meal vouchers per transaction is set at EUR 1.50 on the payment gateway.

The maximum amount is set at EUR 25 per day, that can be paid in one or more installments.

The issuer is responsible for processing immediate debits of account holders' accounts (authorization or refusal) and is solely responsible for refunding merchants.

Titre-Restaurant cards cannot be used on Sundays and on holidays (unless the employer has given a special authorization to his/her employees who work on these days).

This verification is performed by issuers. Therefore, it is not necessary to perform additional verification on the merchant website.

The amount payable by meal vouchers is temporarily blocked until the transaction is finalized or abandoned.

In case the transaction is abandoned or canceled or if the payment session expires, the payment gateway performs a recovery.

This action allows to update the outstanding amount of the holder's card and to display the daily available balance to the buyer.

If the recovery operation fails, after the authorization expires the issuer will automatically adjust the balance of the meal voucher.

2.1. First generation cards

First generation cards, even if they have the logo of the restaurant voucher issuer, have the distinction of being payment cards issued within the VISA or MASTERCARD schemes.

When it comes to authorization, they are processed similarly to Visa or Mastercard cards with a daily limit fixed at EUR 25.

The acceptance of 1st generation cards via the payment gateway requires a Visa or Mastercard e-commerce card acceptance contract (CB or international acquirer of Visa/Mastercard) associated with the shop.

In case a new affiliation contract is opened or terminated with an electronic meal voucher issuer, the merchant must contact E-commerce customer service in order to update the list of card types accepted by the card acceptance contract.

Once the list of accepted issuers is configured for the card acceptance contract, the payment gateway recognizes the meal voucher issuer, regardless of the payment method selected by the buyer, and displays the corresponding logo.

1st generation cards are **compatible with split payment** (also called cascading payment) and allow to pay for a purchase using several cards (1 electronic meal voucher + 1 payment card) via the card acceptance contract if the issuer supports partial authorization.

If the acceptance of meal vouchers for split payments is not configured within the card acceptance contract, the meal vouchers co-badged Visa or Mastercard will be considered as regular VISA or Mastercard cards and will not be used for split payments.

Exception

1st generation Bimpli (ex Apetiz) cards are routed exclusively within the CONECS scheme at their request.

2.2. Second generation cards (CONECS scheme)

For the French market, CONECS, who is the EIG of 4 meal voucher issuers: Bimpli (ex Apetiz), Sodexo, UP and Edenred, has created a technical platform for routing and collecting electronic meal voucher payments via payment terminals and on the Internet.

The cards issued under the CONECS brand are:

- **Bimpli (ex Apetiz)** via Natixis Intertitres
- **Pass Restaurant** via Sodexo
- **Chèque Déjeuner** via Groupe Up
- **Ticket Restaurant** via Edenred

These cards work as three-party scheme cards (the issuer and the acquirer are combined).

The acceptance of 2nd generation cards via the payment gateway requires a **CONECS** acceptance contract associated with the shop.

The acceptance of these cards via the CONECS scheme depends on the affiliation contract that the merchant signed with each issuer and the flow of meal vouchers opened by the issuer via the e-commerce channel.

You can have an acceptance contract with an issuer and not have an open e-commerce channel.

The payment gateway only directs toward CONECS the cards recognized by your CONECS acceptance contract.

Every day, automatic remote configuration allows to retrieve the table of BIN codes (6 first digits of the card) of the issuers authorized for the merchant within the e-commerce channel.

This BIN table affects the card types that will be offered to the buyer within the CONECS scheme.

This table evolves automatically for the e-commerce channel depending on your contracts with issuers, both open and terminated.

If an issuer's BIN code is absent from the table, the card will not be accepted.

2nd generation cards are **compatible with split payment** and allow to pay for a purchase using several cards (1 electronic meal voucher + 1 payment card) if the issuer returns a partial authorization code (10).

2.3. Mixed cards

Mixed cards are cards that can be accepted both within the CONECS scheme (three-party scheme) and the card acceptance network (four-party scheme).

These cards are compatible with split payment:

- if the issuer returns a partial authorization code (10),
- and if the split payment is authorized within the card acceptance contract for the four-party mode.

2.4. Summary

Card type	Acquirer
Bimpli (ex Apetiz) 1G, Conecs 2G and mixed 2G (co-badged VISA)	CONECS
Chèque Déjeuner 1G	Acquirer of the card acceptance contract
Chèque Déjeuner 2G and mixed 2G	CONECS
Pass Restaurant 1G	Acquirer of the card acceptance contract
Pass Restaurant 2G and mixed 2G (co-badged VISA)	CONECS
Titre Restaurant Mastercard Edenred 1G	Acquirer of the card acceptance contract
Titre Restaurant Edenred Conecs 2G	Closed e-commerce flow
Titre Restaurant Edenred mixed 2G (co-badged Mastercard)	Acquirer of the card acceptance contract

Conecs 2G: corresponds to three-party card schemes.

Mixed 2G: corresponds to three-party card schemes if it passes through the Conecs scheme or a four-party card scheme if it passes through the Visa or Mastercard networks.

3. TECHNICAL INFORMATION

Values of the vads_card_brand field	Supported currencies	Supported countries	Authorization validity period	Payment process
APETIZ CHQ_DEJ SODEXO CONECS MASTERCARD	EUR	France	7 days	Deferred capture

Sales channel	
e-commerce	✓
m-commerce	✓
MOTO payment	✗
Interactive Voice Server	✗
Payment order by e-mail/SMS	✓

Operations with transactions	
Cancellation	✓
Refund	✗
Modification	✓
Validation	✓
Duplication	✗
Manual reconciliation	✓

Type of integration	
Redirection	✓
Iframe	✓
JavaScript Client	✓
Data collection form	✓
API Web Services	✓
Mobile SDK	✓
Back Office	✗

Payment type	
Immediate payment	✓
Deferred payment	✓
Payment in installments	✗
Payment by subscription	✗
Payment by file (token or batch)	✓
One-click payment	✓
Payment by wallet	✓

Miscellaneous	
Reporting	✓
Transaction settlement	✗
Chargeback settlement	✗
Extra payment attempts	✓

4. PREREQUISITES

For the merchant

To accept Titre-Restaurant CONECS cards (in the same way as paper meal vouchers), the merchant must:

- contact and obtain an agreement from the French National Commission of Meal Vouchers (Commission Nationale des Titres-Restaurant: <http://www.cntr.fr/V2/home.php>),
- accept the general terms and conditions of affiliation of at least one issuer of meal vouchers mentioned above,
- obtain a technical identifier (IDCONECS), specific to e-commerce and attributed by CONECS,
- sign a card acceptance contract to manage 1st generation cards and the supplement.

Contact E-commerce customer service to request the creation or addition of a CONECS contract and transmit:

- your technical identifier (IDCONECS) specific to e-commerce
- the rank number



CONECS contracts only support payments in euro.

The titres-restaurant are valid for the purchase of any food product, whether or not it is directly consumable.

The eligibility of products is the responsibility of the merchant. The merchant must transmit to the payment gateway the total amount of the shopping cart and a sub-total corresponding to the amount of eligible products payable by electronic meal vouchers.

For example, shipping fees are not payable by meal vouchers, therefore they must be paid separately with an additional CB payment or be offered or included in the price of the meal. This aspect specifically concerns merchants who sell different products, such as large retailers.



The sub-total corresponding to the amount of eligible products payable by Titre-Restaurant cannot be lower than **1.5 euro** (amount defined by the payment gateway). If the value submitted in the form is lower than **1.5 euro**, the logos of different issuers will not be presented on the payment page.

For the buyer

No action required.

The only requirement is the possession of an active, valid and replenished Titre-Restaurant card with, possibly, an additional payment method in case of insufficient balance.

5. PARTIAL CAPTURE AND VALIDATION MODE FOR DRIVE-THROUGH STORES

In case a product is unavailable or if a product amount is different, due to the total weight for example, which would become apparent between the moment when the order is made and the moment when the goods are retrieved, the drive-through store is authorized to decrease the transaction amount.

The merchant can confirm the submission of transactions to be captured manually or automatically.

As long as the transaction has not been captured, the merchant can change the capture date of a transaction using the “*modify*” function (manually or automatically using the **Transaction/Update** Web Service function).

Transactions are submitted to be captured automatically by the payment server in the case of automatic validation, otherwise each transaction must be validated manually using the “*validate*” function (or automatically using the **Transaction/Validate** Web Service function).

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

The validation mode can be customized for every transaction via the parameters transmitted in the payment request.

6. PAYMENT BY TOKEN

Payment by token

The payment by token service allows merchant websites to offer their customers the possibility to make payments by providing their identifier instead of their card details.

Payment by token revolves around two main functions:

- Creating a “token”
- The payment

Creating a token

Upon invitation from the merchant website, the buyer can register their payment method.

The card details are stored in the PCI-DSS environment of the gateway.

This operation is only possible during the payment and only if the amount is higher than €1,5.

Payment by token

After identifying themselves on the merchant website, the buyer is redirected to the payment page and needs to simply confirm their payment by entering the card security code.

Payment by token allows payments to be made using a previously registered token without having to enter the card data. A confirmation step is presented with the transaction summary (number and amount).

Depending on the store configuration, the buyer will have to enter the CVV to validate the payment.

This service requires subscription to an offer that includes payment by token.

7. CASCADING PAYMENT



This service, also called **split payment**, is only available for the integration mode by form with redirection.

Cascading payment is a service allowing the buyer to pay the total purchase amount using several payment methods.

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

The use of a prepaid card with a balance lower than the order amount will automatically result in splitting the payment.

All the transactions of a cascading payment are combined within a payment sequence.

All the transactions of a payment sequence have the same transaction identifier, but a different sequence number (see chapter [Analyzing the payment result](#) on page 46).

8. USING THE FORM WITH REDIRECT

8.1. Offering payment with a Titre-Restaurant card

There are several ways to offer payment by Titre-Restaurant card.

Selection of the payment method on the e-commerce website

This solution is recommended by Titre-Restaurant issuers.

The buyer chooses the type of electronic meal voucher on the merchant website. The buyer's choice is then submitted to the payment gateway via the **vads_payment_cards** field of the form.

Use one of the values below depending on the buyer's choice:

- **APETIZ** to display the BIMPLI (ex APETIZ) payment page
- **EDENRED** to display the EDENRED payment page
- **SODEXO** to display the SODEXO payment page
- **CHQ_DEJ** to display the CHEQUE DEJEUNER payment page

E.g.: **vads_payment_cards="APETIZ"**

If the **vads_payment_cards** field is populated with a single value, and this value corresponds to a type of meal voucher that is not available for the CB and CONECS acceptance contract, the payment form will not be rejected.

The buyer will be redirected to the payment method selection page. All the eligible payment methods (depending on the currency, the minimum or maximum amount, technical constraints such as whether the shopping cart data, the account holder's address and the shipping address are present or not, etc.) associated with the shop will be available.

Payment by Titre Restaurant Apetiz is not available.
Please select another payment method:



Visa



Mastercard



Up Déjeuner



Pass Restaurant

Meal voucher type selection on the payment page

The buyer selects his/her type of electronic meal voucher from the list of payment methods defined by the merchant.

For this, the merchant submits the list of electronic meal vouchers to offer via the **vads_payment_cards** field.

Make up your list depending on the cards that your CONECS acceptance contract accepts for the e-commerce channel, using the values below separated by ";":

- **APETIZ** to display the BIMPLI (ex APETIZ) logo
- **EDENRED** to display the EDENRED logo
- **SODEXO** to display the SODEXO logo
- **CHQ_DEJ** to display the CHEQUE DEJEUNER logo

E.g.: **vads_payment_cards="APETIZ;SODEXO"**

Reminder: issuers are updated by remote configuration automatically and on a daily basis.

Payment method selection on the payment page

This is the default behavior. The buyer chooses his or her payment method from a list defined by the payment gateway, including the available electronic meal voucher.

This solution is not recommended.

To use this solution, the merchant website can:

- submit the **vads_payment_cards** field without a value (empty),
- not send the **vads_payment_cards** field in the form.

Depending on the other parameters submitted in the form (amount, currency, amount eligible in meal vouchers), the payment gateway will display all the eligible payment methods. Depending on the value of the amount that is eligible for meal vouchers, the meal voucher payment buttons will not be provided.

E.g.: a shop that has a CB and a Conecs acceptance contract and authorizes BIMPLI (ex APETIZ) and CHEQUE DEJEUNER payments:

Please select your payment method :



Visa



Mastercard



Bimpli



Pass Restaurant



Up Déjeuner

Selection of the CONECS payment method on the payment page

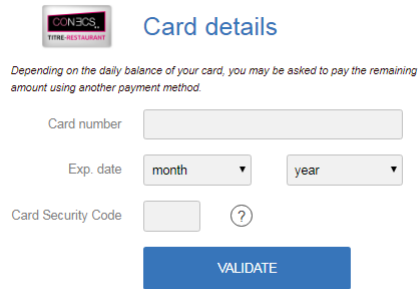
If the merchant does not wish to display the logos of different issuers of the electronic meal voucher, there is a solution that groups all the electronic meal vouchers together.

To use it, the merchant must force the value of the **vads_payment_cards** field in their form with the **"CONECS"** value.

- During payment method selection on the e-commerce website:

vads_payment_cards="CONECS"

The buyer is immediately redirected to the card detail entry page:



The screenshot shows a 'Card details' form. At the top left is a small CONECS logo. To its right is the title 'Card details'. Below the title is a note: 'Depending on the daily balance of your card, you may be asked to pay the remaining amount using another payment method.' The form contains three input fields: 'Card number' (a text box), 'Exp. date' (two dropdown menus for 'month' and 'year'), and 'Card Security Code' (a text box with a question mark icon). A blue 'VALIDATE' button is positioned at the bottom center of the form.

- During payment method selection on the payment page:

Example: **vads_payment_cards="CONECS;CB;VISA;MASTERCARD"**

If the value of the **vads_payment_cards** field is not forced in the form, the payment page will display all the available payment methods with logos of different meal vouchers available with your CB and CONECS acceptance contracts as well as the Conecs logo.

8.2. Integration in the customer journey

In order to simplify the customer journey, increase the conversion rate and thus reduce the number of abandoned orders, it is recommended to:

- select the payment method on the merchant website,
- generate a payment button for each type of payment method.

Note on handling Titre-Restaurant outside the Conecs network.


If you are affiliated with a Titre-Restaurant issuer other than Natixis Intertitres, Sodexo or Groupe Up, you can accept payments with these cards via the payment gateway, but only through your CB acceptance agreement.


This means that:


- Payments made with cards **are not compatible with complementary payments by card**. In other words, you will have to make sure that the amount to be paid does not exceed the daily limit for these cards.
- The eligible amount transmitted via the **vads_acquirer_transient_data** field will not be taken into account. This field is only taken into account for payments made within the Conecs network.
- You can use the MASTERCARD or VISA value. They will allow to apply these meal vouchers, whether they are co-branded Visa or Mastercard.

Examples of payment method selection:


Select your payment method

Pay by Bimpli (ex Apetiz) 

Pay by Chèque-Déjeuner 




Pay by Pass Restaurant 

Others Visa/Mastercard e-meal vouchers


Pay by bank card 

[Pay](#)

Select your payment method:

Pay by Titre-Restaurant   

Others Visa/Mastercard e-meal vouchers

Pay by bank cards 

[Pay](#)

In the example on the left, the choice is between the following values of the **vads_payment_cards** field:

- APETIZ
- SODEXO
- CHQ_DEJ
- MASTERCARD
- CB

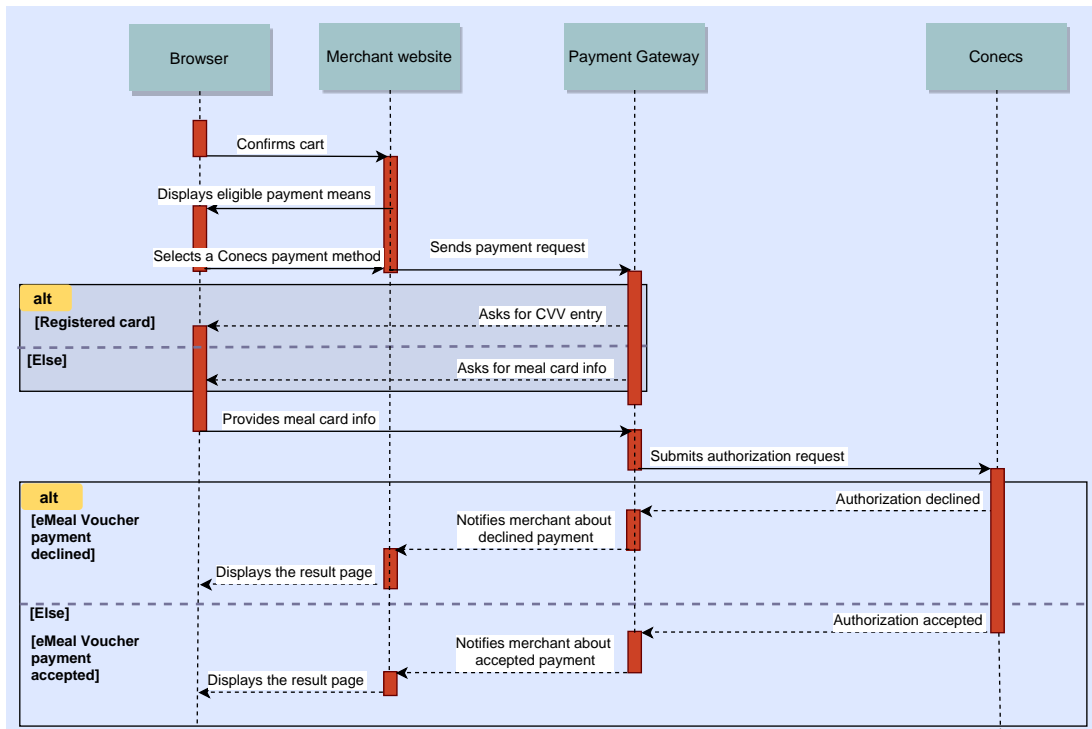
The solution on the right provides a choice between the values:

- CONECS
- MASTERCARD
- CB

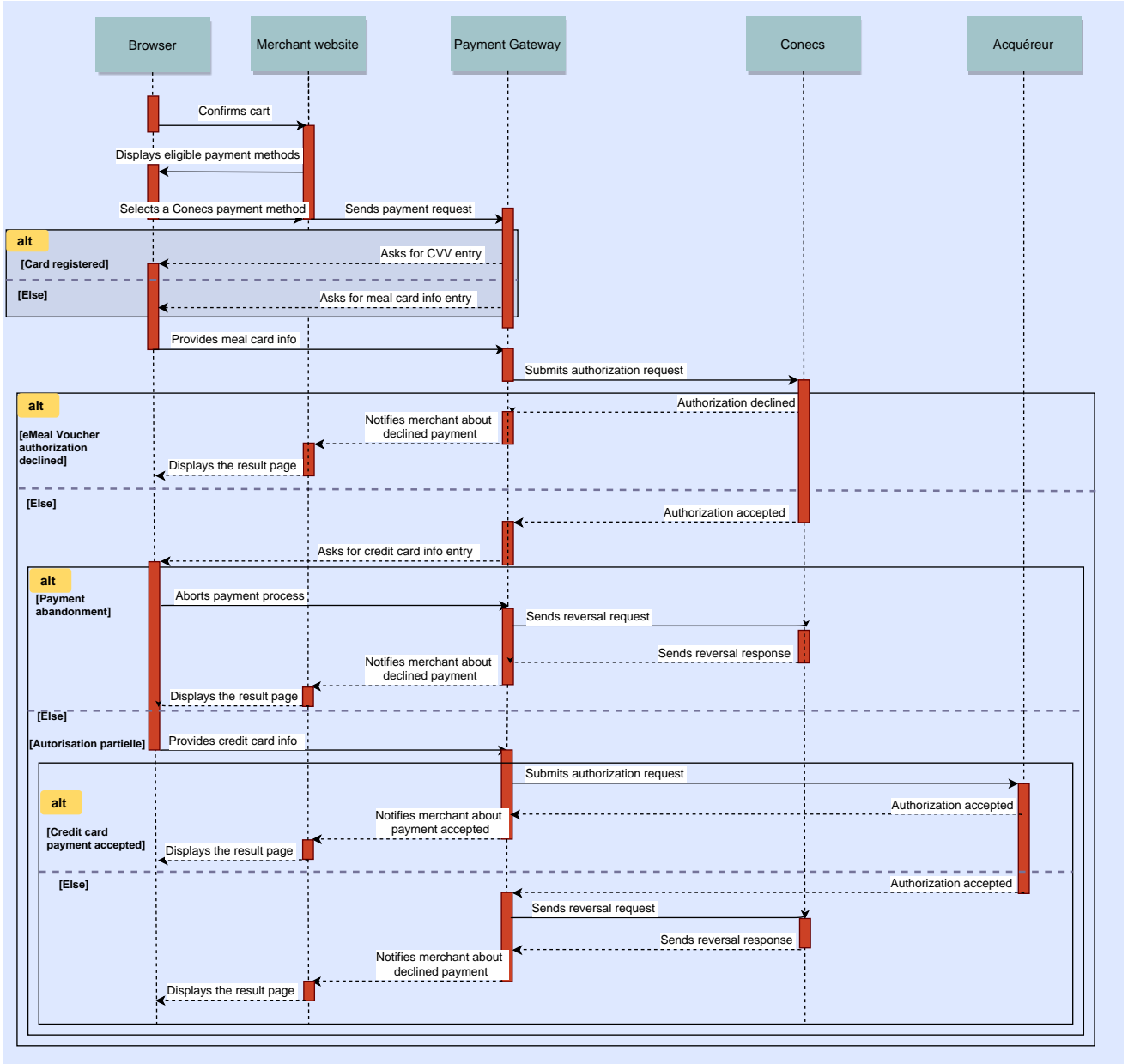
8.3. Understanding the process of a Titre-Restaurant payment

8.3.1. Flow chart

The Titre-Restaurant card allows to pay for 100% of the shopping cart



The Titre-Restaurant card does not allow to pay for 100% of the shopping cart



8.3.2. The process of a single payment

Here is a step-by-step description:

- The payment is made entirely with a Titre-Restaurant card
- The buyer does not have a pre-registered card (not a 1-click payment)
- The buyer does not have the possibility to register their card details

1. The buyer validates the shopping cart.

2. The buyer selects the type of Titre-Restaurant card that will be used.

The card detail entry page appears.

3. The buyer enters their card data.

4. Depending on the shop configuration, the CVV may be requested for validating the payment. In this case, the buyer enters their CVV and clicks **VALIDATE**.

If the entered card number does not correspond to the issuer selected by the buyer, the payment gateway automatically corrects the type before proceeding to payment.

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

Among the displayed information, the logo of the Titre-Restaurant issuer used for the transaction is displayed.

In case of failure, a message is displayed. The buyer is informed of the payment request rejection.

Note: the mobile applications provided by the issuers of the Titre-Restaurant CONECS card allow you to understand the reasons of the payment rejection.

The payment rejection page allows to:

- Return to the shop

Or

- Display the page of payment method selection once again by clicking **New payment attempt**.

This feature is available if the merchant configures it via their Merchant Back Office. For this, via **Settings > Shop > Configuration** tab, the merchant must replace the 0 value specified by default in the field **In case of payment rejection, authorize ... additional attempt(s)** with the desired value (maximum 2 times).

8.3.3. The process of a Titre-Restaurant payment supplemented by another payment method

In this case:

- The buyer does not have a pre-registered card,
- The transaction amount is partly or totally payable by Titre-Restaurant card,
- The buyer uses a Titre-Restaurant card to pay the part payable by Titre-Restaurant,
- The buyer pays the remaining amount with another payment method.

1. The buyer validates the shopping cart.

2. The buyer selects the type of Titre-Restaurant card that will be used.

The card detail entry page appears.

3. The buyer enters their card data.

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

The payment gateway sends an authorization request to retrieve the card balance.

It detects that the card balance is lower than the eligible amount. It prompts to complete the rest of the transaction by bank card.

If the number of the used meal vouchers corresponds to a first generation card, it will be possible to pay the remaining amount with another payment method on the condition that split payment is enabled within the card acceptance contract. Otherwise, the payment will be refused due to insufficient balance.

5. The buyer enters the details of his or her complementary payment method.

The buyer views the amount paid with the Titre-Restaurant card. The buyer can also:

- Delete Titre-Restaurant by clicking the recycling bin icon. This operation will provoke the recovery of Titre-Restaurant via the meal voucher issuer.
- Cancel and return to shop by clicking the corresponding link. This operation will provoke the recovery of Titre-Restaurant via the meal voucher issuer.

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

The complementary payment may generate a 3DS path.

Among the displayed information, the logo of the Titre-Restaurant issuer used for the transaction is displayed.

In case of failure, a message is displayed. The buyer is informed of the payment request rejection.

There are two possibilities:

- If the merchant does not enable the option for additional payment attempts, a message indicating that the payment is rejected will appear. Authorization for Titre-Restaurant is canceled, the balance has been recovered.
- If the merchant enables the option of additional payment attempt, a rejection message is displayed but the payment page prompts the buyer to retry payment. Authorization given for the Titre-Restaurant card is retained until the buyer finalizes or abandons the payment. Only the supplement remains to be paid.

The page of detail entry for the complementary payment method appears.

8.3.4. The process of payment with optional recording (ASK_REGISTER_PAY)

In this case:

- the merchant website prompts the buyer to register the details of his/her electronic meal voucher
- the buyer accepts to register his/her card by checking the corresponding checkbox
- the vads_page_action field is populated with **ASK_REGISTER_PAY** in the payment form
- the merchant has subscribed to an offer that includes payment by token

Payment by token allows to use a pre-registered token for making payments without having to select a payment method and enter the card number.

The beneficiary of a Titre-Restaurant card will no longer have to enter the card number for each transaction. Only the card security code will have to be specified due to security reasons.

In order to offer this option to its buyers, the merchant website must submit a registration request to the payment gateway. The gateway will generate the token value and store the card details.

1. The buyer identifies him/herself on the merchant website.
2. The buyer validates the shopping cart.
3. The buyer selects the logo of his or her Titre-Restaurant's issuer.
The card detail entry page appears.
4. The buyer enters his or her Titre-Restaurant card details (card number, expiration date and security code).
5. The buyer checks the box **I want to register my payment method details for a future purchase**.
6. The buyer clicks **VALIDATE**.
7. There are two possibilities:
 - in case of single payment (without additional payments), the confirmation receipt appears,
 - in case of split payment, the buyer is prompted to complete and proceed with the payment via another payment method.

If the number of the used meal vouchers corresponds to a first generation card, it will be possible to pay the remaining amount with another payment method on the condition that split payment is enabled within the card acceptance contract. Otherwise, the payment will be refused due to insufficient balance. The token will not be created.

At the end of payment, if the buyer has chosen to register his/her card details, the payment gateway will submit the token(s) to the merchant website.

In case of single payment, the meal voucher token will be returned in the **vads_identifier** field.

This token is only valid for meal voucher payments and it will not be possible to use it for a credit card payment.

In order to allow the merchant website to manage the tokens of CB cards and Titre-Restaurant, the response will also contain the type of used card in the **vads_card_brand** field.

The merchant website will have to register the token and its card type and associate them with the buyer's account.

In case of split payment, the meal voucher and the bank card tokens will be returned in the **vads_payment_seq** field. For each transaction, the merchant will have to register the values of the **identifier** and **card_brand** attributes (see chapter [Analyzing the payment result](#) on page 46).

For security reasons, during the next payment, the buyer will have to enter the security code of their meal voucher to validate their payment.

8.3.5. The process of payment with mandatory recording (REGISTER_PAY)

In this case:

- either the merchant website offers the buyer to register the details of their meal voucher and the buyer accepts by checking the box provided for this purpose on the merchant website,
- or the buyer is notified that his/her card details will be registered automatically for facilitating future purchases,
- the vads_page_action field is populated with **REGISTER_PAY** in the payment form,
- the merchant has subscribed to an offer that includes payment by token

1. The buyer identifies him/herself on the merchant website.

2. The buyer validates the shopping cart.

3. The buyer selects the logo of his or her meal voucher's issuer.

The card detail entry page appears.

4. The buyer enters their meal voucher details (card number, expiry date and security code).

5. The buyer clicks **VALIDATE**.

The payment gateway sends an authorization request to retrieve the card balance.

It detects that the card balance is lower than the eligible amount. It prompts to complete the rest of the transaction by bank card.

If the number of the used meal vouchers corresponds to a first generation card, it will be possible to pay the remaining amount with another payment method on the condition that split payment is enabled within the card acceptance contract. Otherwise, the payment will be refused due to insufficient balance. The token will not be created.

6. The buyer enters the details of his or her complementary payment method.

The buyer views the amount paid with the Titre-Restaurant card. The buyer can also:

- Record the payment method details by checking the box **I wish to record my payment method details for a future purchase**.
- Delete Titre-Restaurant by clicking the recycling bin icon. This operation will provoke the recovery of Titre-Restaurant via the meal voucher issuer.
- Cancel and return to shop by clicking the corresponding link. This operation will provoke the recovery of Titre-Restaurant via the meal voucher issuer.

7. The buyer clicks **VALIDATE**.

At the end of payment, if the buyer has chosen to register his/her card details, the payment gateway will submit the token(s) to the merchant website.

In case of single payment, the meal voucher token will be returned in the **vads_identifier** field.

This token is only valid for meal voucher payments and it will not be possible to use it for a credit card payment.

In order to allow the merchant website to manage the tokens of CB cards and Titre-Restaurant, the response will also contain the type of used card in the **vads_card_brand** field.

The merchant website will have to register the token and its card type and associate them with the buyer's account.

In case of split payment, the meal voucher and the credit card tokens will be returned in the **vads_payment_seq** field. For each transaction, the merchant will have to register the values of the **identifiant** and **card_brand** attributes (see chapter [Analyzing the payment result](#) on page 46).

For security reasons, during the next payment, the buyer will have to enter the security code of their meal voucher to validate their payment.

8.3.6. The process of a split payment with a registered card

In this case:

- The buyer uses a pre-registered card (not 1-Click payment).
 - The merchant website submits the token value of the card to be used via the **vads_identifier** field.
1. The buyer validates the shopping cart.
 2. The merchant website redirects the buyer to the payment gateway.
 3. Depending on the shop configuration, the CVV may be requested for validating the payment. In this case, the buyer enters their CVV and clicks **VALIDATE**.
The payment gateway sends an authorization request to retrieve the card balance.
 4. If the balance is lower than the eligible amount, the gateway offers to complete the remaining amount of the transaction by credit card.
 - a. If the buyer has already registered one or more credit cards on the merchant website with the same e-mail address, the payment gateway offers them to choose the card to be used for paying the additional amount.
 - b. Otherwise, the buyer enters the details of their complementary payment method.
The buyer can also:
 - Delete Titre-Restaurant by clicking the recycling bin icon. This operation will provoke the recovery of Titre-Restaurant via the meal voucher issuer.
 - Cancel and return to shop by clicking the corresponding link. This operation will provoke the recovery of Titre-Restaurant via the meal voucher issuer.
 5. In case of successful payment, the summary page is presented to the buyer. In case of failure, a message informs the buyer of payment refusal.

If the number of the used meal vouchers corresponds to a first generation card, it will be possible to pay the remaining amount with another payment method on the condition that split payment is enabled within the card acceptance contract. Otherwise, the payment will be refused due to insufficient balance.

8.4. Generating a payment form

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

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

It contains:

- The following technical elements:
 - The `<form>` and `</form>` tags that allow to create an HTML form.
 - The `method="POST"` attribute that defines the method used for sending data.
 - The `action="https://paiement.systempay.fr/vads-payment/"` attribute that defines where to send the form data.
- Form data:
 - The shop ID.
 - Information about the payment depending on the use case.
 - Additional information depending on your needs.
 - The signature that ensures the integrity of the form.

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

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

For setting the `name` and `value` attributes, see the **Data dictionary** chapter also available in the online document archive.

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

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"/>
```

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

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

Notation	Description
a	Alphabetic characters (from 'A' to 'Z' and from 'a' to 'z')
n	Numeric characters
s	Special characters
an	Alphanumeric characters
ans	Alphanumeric and special characters (except '<' and '>')
3	Fixed length of 3 characters
..12	Variable length up to 12 characters
json	<p>JavaScript Object Notation. Object containing key/value pairs separated by commas. It starts with a left brace "{" and ends with a right brace "}". Each key / value pair contains the name of the key between double-quotes followed by " :", followed by a value. The name of the key must be alphanumeric. The value can be:</p> <ul style="list-style-type: none"> • a chain of characters (in this case it must be framed by double-quotes) • a number • an object • a table • a boolean • empty <p>Example: {"name1":45,"name2":"value2", "name3":false}</p>
bool	Boolean. Can be populated with the true or false value.
enum	<p>Defines a field with a complete list of values. The list of possible values is given in the field definition.</p>
Enum list	<p>List of values separated by a " ; ". The list of possible values is given in the field definition. Example: vads_available_languages=fr;en</p>
map	<p>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:</p> <ul style="list-style-type: none"> • a chain of characters • a boolean • a json object • an xml object <p>The list of possible values for each key/value pair is provided in the field definition. Example: vads_theme_config=SIMPLIFIED_DISPLAY=true;RESPONSIVE_MODEL=Model_1</p>

8.4.1. Transmitting the amount payable by Titre-Restaurant card

The eligibility of products is the responsibility of the merchant. The merchant must transmit to the payment gateway the total amount of the shopping cart and a sub-total corresponding to the amount of eligible products payable by electronic meal vouchers.

The form field with the amount of eligible items payable by Titre-Restaurant card is **vads_acquirer_transient_data**.

This field must respect the following syntax in JSON format (example with an eligible amount of EUR 17.25):

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

The value of the eligible amount must:

- be specified in the smallest currency unit,
- be an integer,
- not be empty,
- be higher or equal to EUR 1.50
- not contain special characters such as full stop, comma or semicolon (".", ",", ";").



The **vads_acquirer_transient_data** field is optional. If the merchant does not transmit it, **the whole amount will be considered eligible for payment by Titre-Restaurant card, including any delivery charges in the order amount.**

If this amount is greater than the daily authorized payment limit, an additional payment method will be required to pay the difference.

8.4.2. Creating a payment request

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

Field name	Description	Format	Value
vads_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.	enum	<ul style="list-style-type: none"> • APETIZ for Bimpli (ex Apetiz) Titres-Restaurant. • SODEXO for Sodexo Titres-Restaurant. • CHQ_DEJ for Chèque-Déjeuner Titres-Restaurant. • CONECS for all Titres-Restaurant processed by the Conecs network. • MASTERCARD for Titres-Restaurant outside the Conecs network.
vads_page_action	Action to perform	enum	PAYMENT
vads_ctx_mode	Mode of interaction with the payment gateway	enum	TEST or PRODUCTION
vads_action_mode	Acquisition mode for payment method data	enum	INTERACTIVE
vads_site_id	Shop ID	n8	E.g.: 12345678
vads_trans_date	Date and time of the payment form in UTC format	n14	Respect the YYYYMMDDHHMMSS format E.g.: 20200101130025
vads_version	Version of the exchange protocol with the payment gateway	enum	V2
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_acquirer_transient_data	Amount payable by Titre-Restaurant card (in the smallest currency unit)	json	Example for an eligible amount of EUR 17.25: vads_acquirer_transient_data = {"CONECS": {"eligibleAmount": "1725"}}
vads_payment_config	Payment type	enum	SINGLE
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_capture_delay	Delay before capture in the bank. Any capture delay that exceeds 7 days will be ignored and reduced to 7 days.	n..3	E.g.: 3
vads_validation_mode	Validation mode	n1	0 (automatic) or 1 (manual)
signature	Signature guaranteeing the integrity of the requests exchanged between the merchant website and the payment gateway.	ans	See Computing the signature on page 41

2. Add the optional fields described in the following chapters according to your requirements.

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

8.4.3. Creating a payment request with registration of an electronic meal voucher

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

Field name	Description	Format	Value
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.	enum	<ul style="list-style-type: none"> • APETIZ for Bimpli (ex Apetiz) Titres-Restaurant. • SODEXO for Sodexo Titres-Restaurant. • CHQ_DEJ for Chèque-Déjeuner Titres-Restaurant. • CONECs for all Titres-Restaurant processed by the Conecs network. • MASTERCARD for Titres-Restaurant outside the Conecs network.
vads_page_action	Action to perform	enum	<ul style="list-style-type: none"> • ASK_REGISTER_PAY Payment with optional registration of the electronic meal voucher card. • REGISTER_PAY Payment with mandatory registration of the electronic meal voucher card.
vads_cust_email	Buyer's e-mail address.	enum	E.g.: name.firstname@example.com
vads_ctx_mode	Operating mode.	enum	TEST or PRODUCTION
vads_action_mode	Acquisition mode for payment method data.	enum	INTERACTIVE
vads_amount	Payment amount (in the smallest currency unit)	n..12	E.g.: 5124 for 51.24 euros
vads_currency	Country code of the card in compliance with the ISO 4217 standard	n3	978 for Euro
vads_acquirer_transient_data	Amount payable by Titre-Restaurant card (in the smallest currency unit)	json	Example for an eligible amount of EUR 17.25: vads_acquirer_transient_data = {"CONECs": {"eligibleAmount": "1725"}}
vads_site_id	Shop ID	n8	E.g.: 12345678
vads_trans_date	Date and time of the payment form in UTC format.	n14	E.g.: 20170501130025
vads_version	Version of the exchange protocol.	string	V2
vads_capture_delay	Delay before capture in the bank. Any capture delay that exceeds 7 days will be ignored and reduced to 7 days.	n..3	E.g.: 3
vads_validation_mode	Validation mode	n1	0 (automatic) or 1 (manual)
signature	Signature guaranteeing the integrity of the requests exchanged between the merchant website and the payment gateway.	ans	See Computing the signature on page 41

2. Add the optional fields described in the following chapters according to your requirements.

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

8.4.4. Creating a payment by token request

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

Field name	Description	Format	Value
vads_page_action	Action to perform	enum	PAYMENT
vads_cust_email	Buyer's e-mail address.	ans..255	E.g.: firstname.name@example.com
vads_identifier	(unique) token associated with a payment method.	enum	E.g.: MyToken
vads_ctx_mode	Operating mode.	enum	TEST or PRODUCTION
vads_action_mode	Acquisition mode for payment method data.	enum	INTERACTIVE
vads_amount	Payment amount (in the smallest currency unit)	n..12	E.g.: 5124 for 51.24 euros
vads_acquirer_transient_data	Amount payable by Titre-Restaurant card (in the smallest currency unit)	json	Example for an eligible amount of EUR 17.25: vads_acquirer_transient_data = {"CONECs": {"eligibleAmount": "1725"}}
vads_currency	Country code of the card in compliance with the ISO 4217 standard	n3	978 for Euro
vads_site_id	Shop ID	n8	E.g.: 12345678
vads_trans_date	Date and time of the payment form in UTC format.	n14	E.g.: 20170501130025
vads_version	Version of the exchange protocol.	string	V2
vads_capture_delay	Delay before capture in the bank. Any capture delay that exceeds 7 days will be ignored and reduced to 7 days.	n..3	E.g.: 3
vads_validation_mode	Validation mode	n1	0 (automatic) or 1 (manual)
signature	Signature guaranteeing the integrity of the requests exchanged between the merchant website and the payment gateway.	ans	See Computing the signature on page 41

2. Add the optional fields described in the following chapters according to your requirements.

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

8.4.5. Creating a payment request with an update of an electronic meal voucher

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

Field name	Description	Format	Value
vads_page_action	Action to perform	enum	REGISTER_UPDATE_PAY
vads_cust_email	Buyer's e-mail address.	ans..255	E.g.: firstname.name@example.com
vads_identifier	(unique) token associated with a payment method.	enum	E.g.: MyToken
vads_ctx_mode	Operating mode.	enum	TEST or PRODUCTION
vads_action_mode	Acquisition mode for payment method data.	enum	INTERACTIVE
vads_amount	Payment amount (in the smallest currency unit)	n..12	E.g.: 5124 for 51.24 euros
vads_acquirer_transient_data	Amount payable by Titre-Restaurant card (in the smallest currency unit)	json	Example for an eligible amount of EUR 17.25: vads_acquirer_transient_data ={"CONECS": {"eligibleAmount":"1725"}}
vads_currency	Country code of the card in compliance with the ISO 4217 standard	n3	978 for Euro
vads_site_id	Shop ID	n8	E.g.: 12345678
vads_trans_date	Date and time of the payment form in UTC format.	n14	E.g.: 20170501130025
vads_version	Version of the exchange protocol.	string	V2
vads_capture_delay	Delay before capture in the bank. Any capture delay that exceeds 7 days will be ignored and reduced to 7 days.	n..3	E.g.: 3
vads_validation_mode	Validation mode	n1	0 (automatic) or 1 (manual)
signature	Signature guaranteeing the integrity of the requests exchanged between the merchant website and the payment gateway.	ans	See Computing the signature on page 41

2. Add the optional fields described in the following chapters according to your requirements.

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

8.4.6. Transmitting buyer details

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

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

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

Field name	Description	Format	Value
vads_cust_email	Buyer's e-mail address	ans..150	E.g.: abc@example.com
vads_cust_id	Buyer reference on the merchant website	an..63	E.g.: C2383333540
vads_cust_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 a private individual COMPANY: for a company
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.

Note: **vads_cust_phone** and **vads_cust_cell_phone** fields accept all formats. Examples:

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

8.4.7. 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

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

Field name	Description	Format	Value
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.).		E.g.: vads_product_ext_id0 = "0123654789123654789" vads_product_ext_id1 = "0223654789123654789"
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 = "Dated 3 days stay" vads_product_label1 = "Private concert"
vads_product_amountN	Item amount expressed in the smallest currency unit. 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 = "32150" vads_product_amount1 = "10700"
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 = "TRAVEL" vads_product_type1 = "ENTERTAINMENT"
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 = "1002127784" vads_product_ref1 = "1002127693"
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 = "1"

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.

8.5. 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 whose name begins with **vads_** alphabetical order.
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 the "+" character.
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 field **vads_ctx_mode**) 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 field **signature**.

Example of parameters sent to the payment gateway:

```
<form method="POST" action="https://paiement.systempay.fr/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. We sort in **alphabetical** order the fields whose name begins with **vads_** :

- 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. We concatenate the value of these fields with 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:
59c96b34c74b9375c332b0b6a32e6deec87de2b

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 field **vads_ctx_mode**) 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=

8.6. Sending the payment request

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

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

8.6.1. Redirecting the buyer to the payment page

The URL of the payment gateway is:

<https://paiement.systempay.fr/vads-payment/>

Example of parameters sent to the payment gateway:

This example shows all the fields that can be sent if this information is available in your information system. Depending on your business needs, certain fields may not be sent.

```
<form method="POST" action="https://paiement.systempay.fr/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_acquirer_transient_data" value="{\"CONECs\":
{\"eligibleAmount\":\"1725\"}}\" />
<input type="hidden" name="vads_amount" value="1915" />
<input type="hidden" name="vads_currency" value="978" />
<input type="hidden" name="vads_cust_id" value="Client" />
<input type="hidden" name="vads_cust_status" value="PRIVATE" />
<input type="hidden" name="vads_cust_first_name" value="Jeanne" />
<input type="hidden" name="vads_cust_last_name" value="Gauthier" />
<input type="hidden" name="vads_cust_phone" value="0123456789" />
<input type="hidden" name="vads_cust_cell_phone" value="0612345678" />
<input type="hidden" name="vads_cust_email" value="jg@sample.com" />
<input type="hidden" name="vads_cust_address_number" value="109" />
<input type="hidden" name="vads_cust_address" value="Rue de l'innovation" />
<input type="hidden" name="vads_cust_zip" value="31670" />
<input type="hidden" name="vads_cust_city" value="Labège" />
<input type="hidden" name="vads_cust_country" value="FR" />
<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" />
<!-- It is recommended sending only one value in the vads_payment_cards field
to prevent the buyer from making an additional choice in the payment page.-->
<input type="hidden" name="vads_payment_cards" value="APETIZ" />
<input type="hidden" name="vads_payment_config" value="SINGLE" />
<input type="hidden" name="vads_nb_products" value="1" />
<input type="hidden" name="vads_product_label0" value="La Végétarienne" />
<input type="hidden" name="vads_product_ref0" value="ZJJ5520" />
<input type="hidden" name="vads_product_qty0" value="1" />
<input type="hidden" name="vads_product_amount0" value="1273" />
<input type="hidden" name="vads_product_type0" value="FOOD_AND_GROCERY" />
<input type="hidden" name="vads_ship_to_type" value="PACKAGE_DELIVERY_COMPANY" />
<input type="hidden" name="vads_ship_to_speed" value="STANDARD" />
<input type="hidden" name="vads_ship_to_delay" value="IMMEDIATE" />
<input type="hidden" name="vads_ship_to_first_name" value="Jeanne" />
<input type="hidden" name="vads_ship_to_last_name" value="Gauthier" />
<input type="hidden" name="vads_ship_to_phone_num" value="0612345678" />
<input type="hidden" name="vads_ship_to_street_number" value="109" />
<input type="hidden" name="vads_ship_to_street" value="Rue de l'innovation" />
<input type="hidden" name="vads_ship_to_zip" value="31670" />
<input type="hidden" name="vads_ship_to_city" value="Labège" />
<input type="hidden" name="vads_ship_to_country" value="FR" />
<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>
```

8.6.2. Processing errors

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

In TEST mode

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

In PRODUCTION mode

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

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

It contains:

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

The e-mail is sent to the company administrator.

If you wish to change this address or add an address, contact E-commerce customer service.

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

To do this:

1. Sign in to your Merchant Back Office:
<https://paiement.systempay.fr/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://paiement.systempay.fr/doc/en-EN/error-code/sitemap.html>

Error codes specific to the CONECS payment method:

Code	Message	Description of the error
130	ACQUIRER_TRANSIENT_DATA	An error will be returned if the entered content is not in JSON format or does not respect the format (CONECS in upper case followed by the = sign) E.g: vads_acquirer_transient_data={"CONECS": {"eligibleAmount": "2058"}}
133	ELIGIBLE_AMOUNT_INVALID	The value of the eligibleAmount element does not respect the expected format: should not contain commas, semicolons and full stops (".", ";", ",;").
134	ELIGIBLE_AMOUNT_NEGATIVE	The value of eligibleAmount cannot be negative (-).
135	ELIGIBLE_AMOUNT_INCONSISTENCY	The value of eligibleAmount cannot be greater than the transaction amount.
136	ELIGIBLE_AMOUNT_MISSING_OR_MISSPELED	eligibleAmount is missing or misspelled.
137	ELIGIBLE_AMOUNT_MIN	The value of the eligibleAmount element cannot be between 0.01 euro and 1.50 euro.
145	NO_CONECS_PAYMENT_CARD	There are no enabled restaurant vouchers associated with your Conecs contract. Contact your Conecs support team.
146	PAYMENT_CARD_CONECS	The vads_payment_cards field is populated with CONECS in the payment form. There are no enabled restaurant vouchers associated with your Conecs contract. Contact your Conecs support team.
147	PAYMENT_CARD_CONECS_APETIZ	The vads_payment_cards field is populated with APETIZ in the payment form. There are no enabled restaurant vouchers associated with your Conecs contract. Contact your Conecs support team.
148	PAYMENT_CARD_CONECS_CHQ_DEJ	The vads_payment_cards field is populated with CHQ_DEJ in the payment form. There are no enabled restaurant vouchers associated with your Conecs contract. Contact your Conecs support team.
149	PAYMENT_CARD_CONECS_EDENRED	The vads_payment_cards field is populated with EDENRED in the payment form. There are no enabled restaurant vouchers associated with your Conecs contract. Contact the restaurant voucher issuer to resolve the issue.
150	PAYMENT_CARD_CONECS_SODEXO	The vads_payment_cards field is populated with SODEXO in the payment form. There are no enabled restaurant vouchers associated with your Conecs contract. Contact your Conecs support team.

Table 1: vads_acquirer_transient_data error codes

8.7. Analyzing the payment result

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

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

8.7.1. Processing the response data

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

1. Identify the order by retrieving the value of the **vads_order_id** field.
Make sure that the order status has not been updated yet.
2. Retrieve the field **vads_trans_uuid** containing the transaction unique reference generated by the payment server.
3. Retrieve the value of the **vads_trans_date** field to identify the payment date.
4. Retrieve the used amount and currency. To do this, retrieve the values of the following fields:

Field name	Description
vads_amount	Payment amount in the smallest currency unit.
vads_currency	Numeric code of the currency used for the payment.

5. Retrieve all the order, buyer and shipping details.
These details will be provided in the response only if they have been transmitted in the payment form.
Their values are identical to the ones submitted in the form.
6. Analyze the value of the **vads_sequence_number** field:

VALUE	DESCRIPTION
0	Case of a cascading payment. Several payment methods were used to pay for the order. The list of payment methods used by the Buyer can be found in the vads_payment_seq field (in the JSON format). For more information, see chapter <i>Analyzing the result of a cascading payment</i> .
1	Order paid entirely with Meal Vouchers.

7. Analyze the value of the **vads_card_brand** field:

Its value allows to know if the order has been paid in full with Titre-Restaurant or if a complementary payment is required.

VALUE	DESCRIPTION
MULTI	Case of a cascading payment. Several payment methods were used to pay for the order. The list of payment methods used by the Buyer can be found in the vads_payment_seq field (in the JSON format). For more information, see chapter <i>Analyzing the result of a cascading payment</i> .
SODEXO	Order paid entirely with "Pass Restaurant" meal vouchers.
EDENRED	Order paid entirely with "Titre Restaurant Mastercard" meal vouchers.
CHQ_DEJ	Order paid entirely with "Chèque Déjeuner" meal vouchers.
APETIZ	Order paid entirely with Bimpli (ex Apetiz) meal vouchers.

8. If the order was paid entirely with Titre-Restaurant, retrieve the payment result transmitted in the **vads_trans_status** field.

Its value allows you to define the order status.

VALUE	DESCRIPTION
AUTHORISED	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 passed, the payment status changes to EXPIRED . The Expired status is final.
REFUSED	Refused Transaction is declined. See the chapter Analyzing the result of the authorization request on page 51 to obtain more information on the reason of payment rejection.
ABANDONED	Abandoned Payment abandoned by the buyer The transaction is visible in the Merchant Back Office.

9. If the order was paid using several payment methods, see the following chapter with the **vads_payment_seq** field analysis.

10. Proceed to order update.

8.7.2. Analyzing the result of a cascading payment

Characteristics of a cascading payment:

When making a payment, the buyer pays a part of the purchase with Titre-Restaurant and the remaining amount with another payment method.

The payment will consist of N sequences detailed in the **vads_payment_seq** field:

- All the transactions of a cascading payment are combined within a payment sequence.
- All the transactions of a payment sequence have the same transaction identifier, but a different sequence number.
- The number of the first transaction of the sequence is 1, the number of the second one is 2, etc.

How to identify a cascading payment?

In case of a cascading payment, the **vads_card_brand** field is set to **MULTI**.

Details of the vads_payment_seq field

All the transactions of a cascading payment are combined within a payment sequence.

All the transactions of a payment sequence have the same transaction identifier, but a different sequence number. The number of the first transaction of the sequence is 1, the number of the second one is 2, etc.

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

- **"trans_id"**: identifier of the transaction received in the **vads_trans_id** field of the payment form.
- **"transactions"**: table of sequence transactions.



canceled and rejected transactions are also displayed in the table.

Each transaction of the sequence 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. <ul style="list-style-type: none"> • For a Titre-Restaurant payment, see the value list in the following chapter. • For payment by bank card, see the value list of the vads_auth_result field in the <i>Hosted Payment Page Implementation Guide</i>. • For other payment methods, see the corresponding integration documentation.
capture_delay	Delay before the capture (in days). If the vads_capture_delay field is not transmitted in the payment request, the delay configured in shop settings will be applied. See the chapter Technical information on page 9 to check if any restrictions apply for this payment method.
card_brand	Code of the used payment method.

Field name	Description												
	<p>Examples:</p> <ul style="list-style-type: none"> • APETIZ for a payment with an Bimpli (ex Apetiz). • SODEXO for a payment with a Sodexo card. • CHQ_DEJ for a payment with a Chèque-Déjeuner card. • VISA for a payment with a Visa card. • MASTERCARD for a payment with a Mastercard card. • CB for a payment with a CB card. <p>See <i>Hosted Payment Page Implementation Guide</i> to view the full list of vads_card_brand field values.</p>												
card_number	Payment method number or ID of the account used for the payment.												
expiry_month	Expiry month of the payment method. Will not be returned if not applicable to the used payment method.												
expiry_year	Expiry year of the payment method. Will not be returned if not applicable to the used payment method.												
payment_certificate	Payment certificate. Will not be returned if not applicable to the used payment method.												
contract_used	Contract number used for the payment.												
identifier	Token associated with a payment method. Will not be returned if not applicable to the used payment method.												
identifier_status	<p>Only present if the requested action is a token creation or update. Possible values:</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>CREATED</td> <td>The authorization request has been accepted. The token has been successfully created.</td> </tr> <tr> <td>NOT_CREATED</td> <td>The authorization request has been declined. The token has not been created, and therefore cannot be viewed in the Merchant Back Office.</td> </tr> <tr> <td>UPDATED</td> <td>The token has been successfully updated.</td> </tr> <tr> <td>NOT_UPDATED</td> <td>The token has not been updated.</td> </tr> <tr> <td>ABANDONED</td> <td>The action has been abandoned by the buyer (debtor). The token has not been created, and therefore cannot be viewed in the Merchant Back Office.</td> </tr> </tbody> </table>	Value	Description	CREATED	The authorization request has been accepted. The token has been successfully created.	NOT_CREATED	The authorization request has been declined. The token has not been created, and therefore cannot be viewed in the Merchant Back Office.	UPDATED	The token has been successfully updated.	NOT_UPDATED	The token has not been updated.	ABANDONED	The action has been abandoned by the buyer (debtor). The token has not been created, and therefore cannot be viewed in the Merchant Back Office.
Value	Description												
CREATED	The authorization request has been accepted. The token has been successfully created.												
NOT_CREATED	The authorization request has been declined. The token has not been created, and therefore cannot be viewed in the Merchant Back Office.												
UPDATED	The token has been successfully updated.												
NOT_UPDATED	The token has not been updated.												
ABANDONED	The action has been abandoned by the buyer (debtor). The token has not been created, and therefore cannot be viewed in the Merchant Back Office.												
presentation_date	Desired date of capture at the bank (in ISO-8601 format). Example: 2020-04-08T14:56:48Z												
trans_id	Transaction number.												
ext_trans_id	Transaction reference on the acquirer side. Will not be returned if not applicable to the used payment method.												
trans_uuid	Unique reference generated by the payment gateway after the creation of a payment transaction. Guarantees that each transaction is unique.												
sequence_number	Sequence number.												
trans_status	Status of the transaction.												

Field name	Description
	<ul style="list-style-type: none"> For a Conecs payment, see the chapter <i>Processing the response data</i> on page 46 to obtain the list of possible statuses. For payment by bank card, see the value list of the vads_trans_status field in the <i>Hosted Payment Page Implementation Guide</i>. For other payment methods, see the corresponding integration documentation.
extra_result	Numeric code of the risk assessment result. Will not be returned if not applicable to the used payment method. See <i>Hosted Payment Page Implementation Guide</i> to view the full list of vads_extra_result field values.

8.7.3. Analyzing the result of the authorization request

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

For a successful Titre-Restaurant card payment, the **vads_auth_result** field can be populated in two ways:

- **00**: The payment has been accepted.
- **10**: Payment by meal voucher has been accepted, a complementary payment by payment card has been requested.

A rejected payment by Titre-Restaurant returns the table of return codes below.

Only the card issuer can provide the real reason of the return code.

The payment gateway only retranscribes this return code without modifying it.

For more information, the buyer must contact his or her issuer.

Codes	Description	Comment
03	Invalid acceptor	A non-exhaustive list of possible cases: <ul style="list-style-type: none"> • The CONECS (IDCONECS) technical identifier is unknown to the CONECS gateway • The CONECS (IDCONECS) technical identifier is not open on the Titres-Restaurant issuer side
05	Do not honor	Several possible causes. Request the verification of the MCC linked to the card acceptance contract of your shop.
14	Invalid cardholder number	A non-exhaustive list of possible cases: <ul style="list-style-type: none"> • invalid card number • invalid expiration date
30	Format error	
56	Card absent from the file	
57	Transaction not allowed for this cardholder	A non-exhaustive list of possible cases: <ul style="list-style-type: none"> • the contract does not support the card type used by the buyer • insufficient funds • the card was used on a Sunday or a holiday • blocked card • rejected payment • other possible cases
59	Suspected fraud	A non-exhaustive list of possible cases: <ul style="list-style-type: none"> • fraud alert raised by the card issuer • other possible cases
63	Security rules unfulfilled	
82	Incorrect CVV	
90	Temporary shutdown	Temporary shutdown
91	Inaccessible card issuer	
94	Duplicated transaction	Suspected duplicated transaction
96	System malfunction	System malfunction
97	Global surveillance timeout deadline	
98	Server not available, new network route requested	
99	Initiator domain incident	System malfunction

9. USING THE JAVASCRIPT CLIENT (EMBEDDED FORM)

9.1. Operating principle



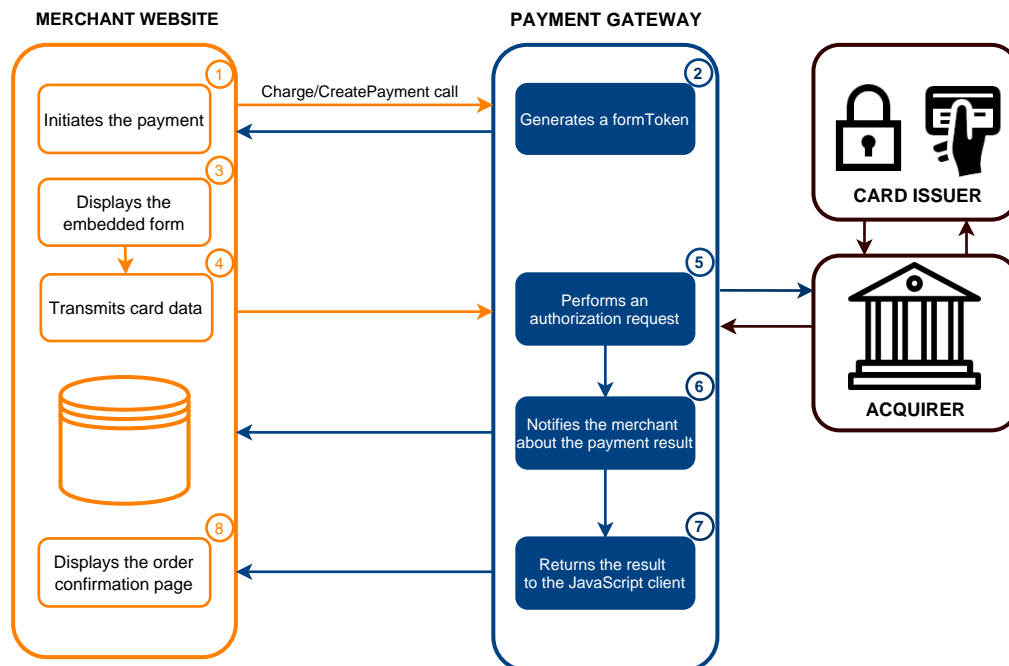
This integration mode is reserved for merchants exclusively selling products eligible for payment by meal vouchers.

Moreover, it is not possible to settle an order using several payment methods (e.g. no CB complement).

Using the **acquirerTransientData** field for specifying the eligible amount will cause the request to be rejected (INT_865 code).

If the amount to be paid is higher than the daily limit or the legal limit in force, the payment will be refused.

For more information on how to integrate the JavaScript client on your website, see the documentation https://paiement.systempay.fr/doc/en-EN/rest/v4.0/javascript/quick_start_js.html



1. The merchant website initiates a payment request via a call to the **Charge/CreatePayment** Web Service.
2. The payment gateway returns a **formToken** to the merchant website.
3. The merchant website uses the **formToken** to display the embedded form.
4. The buyer enters their meal voucher number and confirms the entry.
The card details are transmitted by the JavaScript client to the payment gateway.
5. The payment gateway performs an authorization request.
If the issuer accepts the request, the payment is accepted.
If the payment is refused, the buyer is invited to try to make another payment.

6. The payment gateway notifies the merchant website.
7. The payment gateway sends the response to the JavaScript client.
8. The merchant website displays the order confirmation page.

9.2. Initializing a single payment request

The merchant website initiates a payment request via a call to the **Charge/CreatePayment** Web Service.

POST <https://api.systempay.fr/api-payment/V4/Charge/CreatePayment>

In response, the merchant server retrieves a **formToken**, an encrypted object allowing to initialize the embedded form with the transaction details and the details corresponding to the shop configuration.

Use the fields below to build your request:

Field name	Description	Required
formAction	Type of desired behavior when a transaction is being created. Must be set to PAYMENT .	No
amount	Amount to be paid, expressed in the smallest currency unit (cents for euro).	Yes
currency	Payment currency. ISO 4217 alpha-3 code. E.g.: "EUR" for euro	Yes
orderId	Order reference. This field is recommended.	No
customer.email	E-mail address that will be used for sending the payment receipt to the buyer. This field is recommended.	No
customer.reference	Buyer ID on the merchant website. This field is recommended.	No
customer.billingDetails	Object containing details of the billing address. This field and its attributes are recommended.	No
customer.shippingDetails	Object containing the details of the billing address. This field and its attributes are recommended.	No
customer.shoppingCart	Object containing the details of the shopping cart. This field and its attributes are recommended.	No
transactionOptions.cardOptions.manualValidation	Payment validation mode.	No
transactionOptions.cardOptions.captureDelay	Capture delay. Any capture delay that exceeds 7 days will be ignored and reduced to 7 days.	No



In order to obtain a more detailed description of the fields to use, test the **Charge/CreatePayment** Web Service using our [playground](#).

Example of a request:

POST <https://api.systempay.fr/api-payment/V4/Charge/CreatePayment>

```
{
  "amount": 1715,
  "currency": "EUR",
  "formAction": "PAYMENT",
  "orderId": "CMD012859"
  "customer": {
    "reference": "104123487",
    "email": "sample@example.com",
    "billingDetails": {
      "category": "PRIVATE",
      "firstName": "Jeanne",
      "lastName": "Gauthier",
      "phoneNumber": "0123456789",
      "cellPhoneNumber": "0612345678",
      "streetNumber": "109",
      "address": "Rue de l'innovation",
      "zipCode": "31670",
      "city": "Labège",
      "country": "FR",
      "language": "fr"
    },
    "shoppingCart": {
      "cartItemInfo": [
        {
          "productLabel": "La Végétarienne",
          "productType": "FOOD_AND_GROCERY",
          "productRef": "ZJJ5520",
          "productQty": 1,
          "productAmount": "1273"
        }
      ]
    },
    "shippingDetails": {
      "firstName": "Jeanne",
      "lastName": "Gauthier",
      "phoneNumber": "0612345678",
      "address": "Rue de l'innovation",
      "streetNumber": "109",
      "zipCode": "31670",
      "city": "Labège",
      "country": "FR",
      "shippingMethod": "PACKAGE_DELIVERY_COMPANY"
    }
  }
}
```

Example of a response

```
{
  "webService": "Charge/CreatePayment",
  "version": "V4",
  "applicationVersion": "5.25.0",
  "status": "SUCCESS",
  "answer": {
    "formToken": "02iz0LcBypiLCJWSVNBX0RFQklUIiwiVklTQV...",
    "_type": "V4/Charge/PaymentForm"
  },
  "ticket": "111a64d6eb464bb8841f1c325af503af",
  "serverDate": "2021-06-29T12:37:33+00:00",
  "applicationProvider": "NPS",
  "metadata": null,
  "mode": "TEST",
  "serverUrl": "https://api.systempay.fr/",
  "_type": "V4/WebService/Response"
}
```

9.3. Registering a meal voucher



The Conecs meal voucher can only be registered during a payment.

The **Charge/CreateToken** Web Service is not supported.

For first generation meal vouchers, or those using the CB contract, the use of the CreateToken Web Service is possible.

To request the registration of a meal voucher during the payment, call the **Charge/CreatePayment** Web Service.

POST <https://api.systempay.fr/api-payment/V4/Charge/CreatePayment>

Use the same fields as for a single payment and change the value of the **formAction** field using one of the values below:

Values of the formAction field	Description
REGISTER_PAY	Forces the registration of the meal voucher during the payment.
ASK_REGISTER_PAY	Proposes the registration of the meal voucher during the payment by means of a checkbox.

Example of a request:

```
POST https://api.systempay.fr/api-payment/V4/Charge/CreatePayment
```

```
{
  "amount": 1715,
  "currency": "EUR",
  "formAction": "ASK_REGISTER_PAY",
  "orderId": "CMD46478",
  "customer": {
    "email": "sample@example.com",
    "reference": "104123487",
    "billingDetails": {
      "category": "PRIVATE",
      "firstName": "Jeanne",
    }
  }
  ...
}
```

If the payment is accepted, a payment method token is created. The token is returned in the **paymentMethodToken** field of the response transmitted by the buyer's browser and in the IPN.

9.4. Using a registered payment method

Call the **Charge/CreatePayment** Web Service:

POST <https://api.systempay.fr/api-payment/V4/Charge/CreatePayment>

Use the same fields as for a single payment and add the **paymentMethodToken** field:

Field name	Description
paymentMethodToken	Token of the payment method as returned after the meal voucher registration.

Note on the buyer details associated with the payment method token

During a payment by token, the attributes **customer.reference**, **customer.email** and **customer.billingDetails** transmitted in the request are ignored and the values associated with the token are used.

Example of a request:

```
POST https://api.systempay.fr/api-payment/V4/Charge/CreatePayment
```

```
{
  "amount": 1715,
  "currency": "EUR",
  "formAction": "PAYMENT",
  "orderId": "CMD46478",
  "paymentMethodToken": "951dabcc0ebf4070a5d6709ad96e82ad",
  "customer": {
    "shippingDetails": {
      "category": "PRIVATE",
      "firstName": "Jeanne",
      ...
    }
  }
}
```

Once the form is displayed, the payment method details will be automatically pre-filled.

All the buyer needs to do is validate the payment (depending on the shop configuration, the CVV may be requested for validating the payment).

9.5. Using the buyer wallet



Payment by meal voucher is compatible with the use of the buyer wallet.
For more information on the buyer wallet, see our integration guide.

In order to use the buyer wallet, make a call to the **Charge/CreatePayment** Web Service.

POST <https://api.systempay.fr/api-payment/V4/Charge/CreatePayment>

Use the same fields as for a single payment and change the value of the **formAction** field using the value below:

Value of the formAction field	Description
CUSTOMER_WALLET	Proposes to add the payment method in the buyer wallet and allows to use already registered payment methods. The customer.reference field becomes mandatory.

Note on the buyer details associated with the buyer wallet

The billing information (**billingDetails** object) transmitted during the registration of the payment method is automatically applied to the transactions made with the buyer wallet.

However, if the merchant transmits the billing details in the **Charge/CreatePayment** request, then it is the request data that is used for the transaction.

In this case, the buyer details associated with the payment method are not updated.

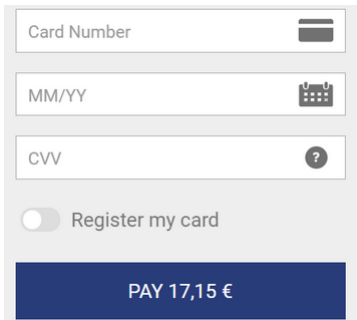
Example of a request:


POST <https://api.systempay.fr/api-payment/V4/Charge/CreatePayment>


```
{
  "amount": 1715,
  "currency": "EUR",
  "formAction": "CUSTOMER_WALLET",
  "customer": {
    "reference": "104123487",
    "email": "sample@example.com",
    "billingDetails": {
      "category": "PRIVATE",
      "firstName": "Jeanne",
      ...
    }
  }
}
```


Examples of a form

If no meal vouchers are associated with the buyer wallet, the form will propose to enter the card details:



Card Number 

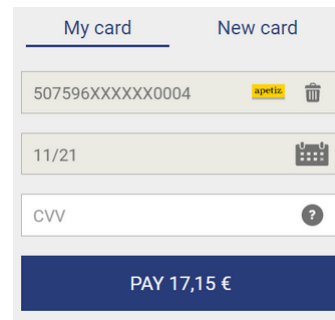
MM/YY 

CVV 



Register my card


PAY 17,15 €


If a meal voucher is associated with a buyer wallet, the fields are pre-filled. Th buyer will also have the possibility to add a new card:



My card **New card**

507596XXXXXX0004  

11/21 

CVV 

PAY 17,15 €

9.6. Analyzing the response

<p>orderStatus Description: Simplified transaction status. Possible values:</p> <ul style="list-style-type: none"> • PAID: The order has been paid. • UNPAID: The order has not been paid. • RUNNING: The order is in progress. <p>E.g.: PAID</p>
<p>orderCycle Description: Define if the order is open (OPEN, the payment is authorized), or closed (CLOSED, it can no longer be modified). Possible values:</p> <ul style="list-style-type: none"> • OPEN: The order is in progress. You must store the cart as there may be further payment attempts. • CLOSED: The order is either fully paid or the last attempt was refused (in which case you must generate a new formToken). <p>E.g.: CLOSED</p>
<p>shopId Description: Shop ID E.g.: 12345678</p>
<p>orderDetails.mode Description: Transaction creation mode. Possible values:</p> <ul style="list-style-type: none"> • TEST: for a test order • PRODUCTION: for a real order <p>E.g.: PRODUCTION</p>
<p>orderDetails.orderId Description: Order reference. E.g.: myOrderId-1234</p>
<p>customer.reference Description: Buyer ID on the merchant website. E.g.: MyReference-123456</p>
<p>customer.email Description: Buyer's e-mail address. E.g.: sample@example.com</p>
<p>cusotmer.billingDetails Description: Object containing details of the billing address.</p>
<p>customer.shippingDetails Description: Object containing the details of the billing address.</p>
<p>customer.shoppingCart Description: Object containing the details of the shopping cart.</p>
<p>transaction.paymentMethodToken Description: Token of the payment method. E.g.: 7d7eeb9794ff473e8843c3ef4bb690b5</p>
<p>transaction.amount Description: Payment amount in the smallest currency unit (cents for euro). E.g.: 4525 for EUR 45.25</p>
<p>transaction.currency Description: Payment currency according to the ISO 4217 alpha-3 standard. Value: Always set to "EUR".</p>
<p>transactions.transactionDetails.acquirerNetwork Description: Acquirer network code. Possible values:</p> <ul style="list-style-type: none"> • CONECS if the payment is made with the Conecs contract. • CB if the payment is made with he CB contract (e.g., in case of first-generation meal vouchers).

<p>E.g.: CONECS</p> <p>transactions.transactionDetails.cardDetails.effectiveBrand</p> <p>Description: Card brand.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • APETIZ • CHQ_DEJ • SODEXO • MASTERCARD <p>E.g.: APETIZ</p>
<p>transactions.uuid</p> <p>Description: Unique transaction identifier generated by the payment gateway.</p> <p>This identifier is then used for transaction management using the REST Web Service or via the Back Office.</p> <p>E.g.: aab72593d0ce428ca6dd73ac7293d92d</p>
<p>transactions.transactionDetails.cardDetails.legacyTransId</p> <p>Description: Unique transaction identifier generated by the payment gateway.</p> <p>This identifier is then used for transaction management via the Merchant Back Office.</p> <p>E.g.: 992364</p>
<p>transactions.transactionDetails.cardDetails.manualValidation</p> <p>Description: Transaction validation mode.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • NO: Automatic validation by the payment gateway. • YES: Manual validation by the merchant. <p>E.g.: NO</p>
<p>transactions.transactionDetails.cardDetails.expectedCaptureDate</p> <p>Description: Date of capture at the bank expressed in ISO 8601 format.</p> <p>E.g.: 2021-09-09T07:46:06+00:00</p>
<p>transactions.transactionDetails.cardDetails.authorizationResponse.authorizationResult</p> <p>Description: Return code of the authorization request.</p> <p>E.g.: 00</p>

10. MANAGING CONECS TRANSACTIONS FROM THE MERCHANT BACK OFFICE

10.1. Viewing transaction details

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

- A Titre-Restaurant transaction that is accepted but not captured can be viewed in the **Transactions is progress** tab until its capture.
- A Titre-Restaurant transaction that is captured can be found in the **Captured transactions** tab.
- A complementary payment method transaction, i.e. by credit card, can be viewed in the **Transactions is progress** tab until its capture.

Note: a Titre-Restaurant transaction that is canceled or refused can be found in the **Transactions is progress** tab.

To view the details of a Titre-Restaurant transaction:

1. Select **Management > Transactions** menu.

By default, the **Transactions is progress** tab appears.

2. Select the **Captured transactions** tab to view the list of captured transactions.

OR

Select the **Transactions is progress** tab to view the list of transactions of the day that have not been captured yet.

3. Select a Titre Restaurant transaction.

4. Right click on it and select **Display order details** or double-click the transaction you wish to see the details of.

Details of a transaction in progress: 091544
✕

Details

Split payment
Buyer
Extra
Risk assessment
Event log

Transaction identification

Transaction :	091544
Transaction UUID :	39a8d570bb874ca39cad3cc72dda03b8
Order reference :	123456789
Sequence number :	1
Shop :	
Current amount :	EUR 19.00
Type :	Debit

Transaction life cycle

Status :	Waiting for capture
Creation date :	11/06/2018 11:15:46
Requested capture date :	11/06/2018 11:15:46

Payment method

Payment method :	☒
Card number :	507596XXXXXX0046 (2019/06 - valid)

Authorization

Merchant ID (MID) :	1999011
Terminal ID (TID) :	001
Pre-authorization :	None
Authorization number :	3fe23d
Authorization return :	10: Partially approved
Authorization date :	11/06/2018 11:15:46
Eligible amount in e-Meal Vouchers :	EUR 30,00

Validate | Modify | Cancel | Duplicate | Receipt

✕ Close

Among the presented information, you will find:

Details tab

- the transaction identifier
- the sequence number
- the total amount
- the amount payable by Titre-Restaurant card
- the used payment method (type of card used)
- the return code of the authorization

Example: Code 10: Partially approved in the details of a Titre-Restaurant transaction

Split payment tab

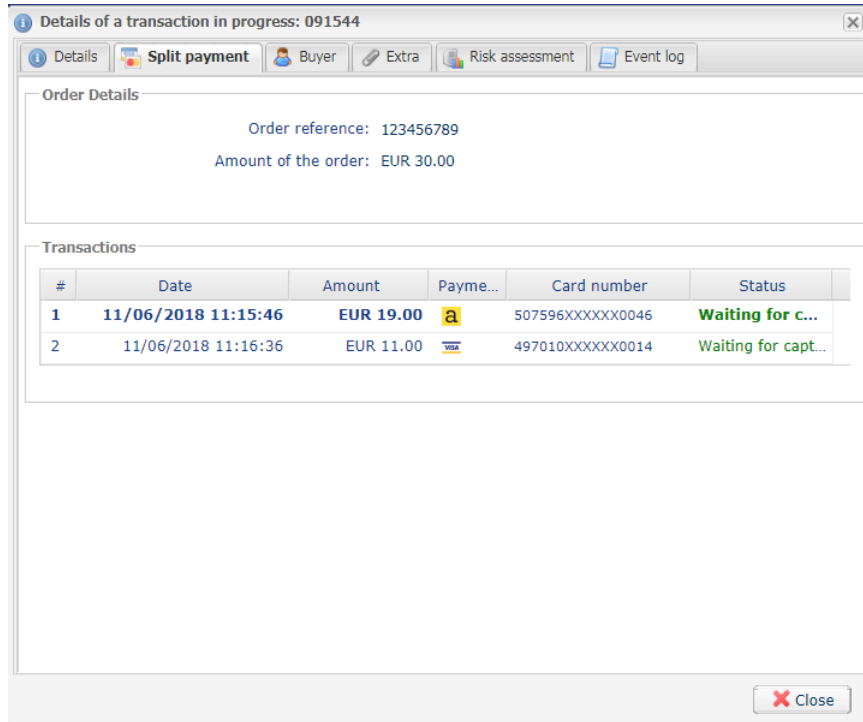
- The sequences of the split payment (Titre-Restaurant card and bank card).

Note:

All payment sequences are recorded: canceled payment, rejected payment, payment waiting for capture.

Each sequence can be selected with a double-click for more details.

- The status of each payment sequence

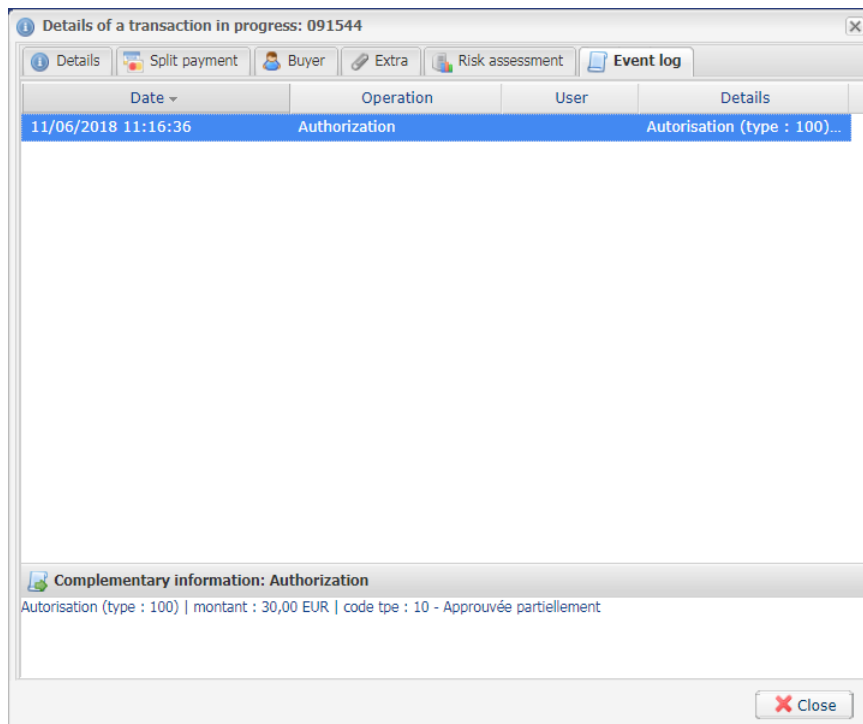


History tab

The **History** tab presents all the operations of the selected transaction.

Each of the operations can be selected to see its details.

- In case of cancellation, the recovery action performed by the payment gateway is specified in the **Info** column. The details (code and title) of this operation can be found at the bottom of the window in the **Additional information** section.
- Capture at the bank



10.2. Canceling a transaction

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

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

The transaction status changes to **Canceled**.

Important:

Canceling meal voucher payment sequence only cancels the selected sequence. The other sequences of the transaction remain waiting for capture. To cancel a transaction that consists of several sequences, the **Cancel** action must be performed for all sequences of a payment. Otherwise, one part of the transaction (e.g.: bank card) will be captured.

Note:

All cancellations result in the recovery of the meal voucher.

Note

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

*For this, select all the transactions to be canceled. Press and hold down the **Ctrl key** and **click** for selecting multiple transactions.*

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

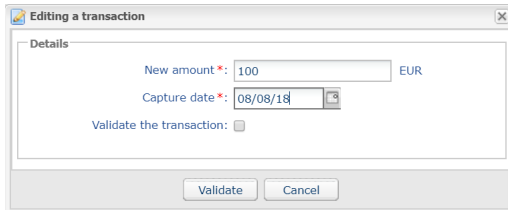
*The transaction statuses will change to **Canceled**.*

10.3. Modifying a transaction

The **Modify** option is only available as long as the capture date has not been reached.

To modify a transaction:

1. Right-click the transaction
2. Select **Modify**.



3. Enter a new amount.

The new amount must be lower than the initial amount.

4. Specify the capture date.

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

5. Click **Validate**.

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

10.4. Validating a transaction

This operation allows to indicate that the transaction can be captured on the scheduled presentation date.

Only the transactions with the following statuses can be validated:

- **To be validated**
- **To be validated and authorized**

In order to validate a transaction:

1. Click on the **Transactions in progress** tab.
2. Select the transaction.
3. Click **Validate**.

Once the transaction has been validated, the status changes to “**Waiting for capture**” or “**Waiting for authorization**” depending on the initial transaction status.

Even if it is not validated before the scheduled capture date, the payment status will remain To be validated until the authorization expires.

In the meantime, you will still be able to validate and/or modify it even if the initial capture date has passed.

11. MANAGING TRANSACTIONS USING REST WEB SERVICES

11.1. Retrieving transaction details

In order to retrieve transaction details, make a call to the **Transaction/Get** Web Service.

POST <https://api.systempay.fr/api-payment/V4/Transaction/Get>

Use the fields below to build your request:

Field name	Description	Required
uuid	Unique transaction identifier generated by the payment gateway.	Yes

Example of a request:

```
POST https://api.systempay.fr/api-payment/V4/Transaction/Get
```

```
{  
  "uuid": "a0b61d1e159748eaab876cfe342e8c08"  
}
```

The Web Service returns a **Transaction** object containing the payment details.

In order to obtain more information on the contents of the **Transaction** object, test the **Transaction/Get** Web Service using our [playground](#).

11.2. Canceling a transaction

In order to cancel a transaction, make a call to the **Transaction/CancelOrRefund** Web Service.

POST <https://api.systempay.fr/api-payment/V4/Transaction/CancelOrRefund>

Use the fields below to build your request:

Field name	Description	Required
uuid	Unique transaction identifier generated by the payment gateway.	Yes



Warning, this Web Service allows to cancel or refund a payment. However, the refund is not supported by payments via meal vouchers.

Example of a request:

```
POST https://api.systempay.fr/api-payment/V4/Transaction/CancelOrRefund
```

```
{  
  "uuid": "a0b61d1e159748eaab876cfe342e8c08"  
}
```

The Web Service returns a **Transaction** object containing the payment details.

In order to obtain more information on the contents of the **Transaction** object, test the **Transaction/CancelOrRefund** Web Service using our [playground](#).

11.3. Modifying a transaction

In order to modify a transaction, make a call to the **Transaction/Update** Web Service.

POST <https://api.systempay.fr/api-payment/V4/Transaction/Update>

Use the fields below to build your request:

Field name	Description	Required
uuid	Unique transaction identifier generated by the payment gateway.	Yes
orderId	Order reference.	No
comment	Comment.	No
cardUpdate.amount	Payment amount in the smallest currency unit.	Yes
cardUpdate.currency	Alphabetic code of the currency (ISO 4217 alpha-3).	Yes
cardUpdate.expectedC	Date of capture at the bank expressed in ISO 8601 format. E.g.: 2021-09-09T07:46:06+00:00	No
cardUpdate.manualVal	Transaction validation mode. <ul style="list-style-type: none">NO: Automatic validation by the payment gateway.YES: Manual validation by the merchant.	No

Example of a request:

```
POST https://api.systempay.fr/api-payment/V4/Transaction/Update
```

```
{  
  "uuid": "a0b61d1e159748eaab876cfe342e8c08"  
}
```

The Web Service returns a response of **Common/ResponseCodeAnswer** type.

In order to obtain more information on the contents of the **Transaction** object, test the **Transaction/Update** Web Service using our [playground](#).

11.4. Validating a transaction

In order to validate a transaction, make a call to the **Transaction/Validate** Web Service.

POST <https://api.systempay.fr/api-payment/V4/Transaction/Validate>

Use the fields below to build your request:

Field name	Description	Required
uuid	Unique transaction identifier generated by the payment gateway.	Yes
comment	Comment.	No

Example of a request:

```
POST https://api.systempay.fr/api-payment/V4/Transaction/Validate
```

```
{  
  "uuid": "a0b61d1e159748eaab876cfe342e8c08"  
}
```

The Web Service returns a **Transaction** object containing the payment details.

In order to obtain more information on the contents of the **Transaction** object, test the **Transaction/Validate** Web Service using our [playground](#).

11.5. Zero-click payment (transaction initiated by the merchant)



This integration mode is reserved for merchants exclusively selling products eligible for payment by meal vouchers, in canteens or intelligent vending fridges.

The payment is made from server to server, without any interaction with the buyer.

If the amount to be paid is higher than the daily limit of the card or the legal limit in force, the payment will be refused.

The merchant website initiates a payment request via a call to the **Charge/CreatePayment** Web Service in SILENT mode.

POST <https://api.systempay.fr/api-payment/V4/Charge/CreatePayment>

Use the fields below to build your request:

Field name	Description	Required
formAction	Type of desired behavior when a transaction is being created. Must be set to SILENT .	YES
amount	Amount to be paid, expressed in the smallest currency unit (cents for euro).	Yes
currency	Payment currency. ISO 4217 alpha-3 code. E.g.: "EUR" for euro	Yes
orderId	Order reference. This field is recommended.	No
paymentMethodToken	Token of the payment method as returned after the meal voucher registration.	Yes
customer.shippingDetails	Object containing the details of the billing address. This field and its attributes are recommended.	No
customer.shoppingCart	Object containing the details of the shopping cart. This field and its attributes are recommended.	No
transactionOptions.cardOptions.manualValidation	Payment validation mode.	No
transactionOptions.cardOptions.captureDelay	Capture delay. Any capture delay that exceeds 7 days will be ignored and reduced to 7 days.	No

Note on the buyer details associated with the payment method token

During a payment by token, the attributes **customer.reference**, **customer.email** and **customer.billingDetails** transmitted in the request are ignored and the values associated with the token are used.

The Web Service returns a **Transaction** object containing the payment details.

In order to obtain more information on the contents of the **Transaction** object, test the **Charge/CreatePayment** Web Service in SILENT mode using our [playground](#).

Example of a request:

```
POST https://api.systempay.fr/api-payment/V4/Charge/CreatePayment
```

```
{
  "amount": 1715,
  "currency": "EUR",
```



```
"formAction": "SILENT",  
"orderId": "CMD46478",  
"paymentMethodToken": "951dabcc0ebf4070a5d6709ad96e82ad",  
"customer": {  
  "shippingDetails": {  
    "category": "PRIVATE",  
    "firstName": "Jeanne",  
    ...  
  }  
}
```

12. OBTAINING HELP

Looking for help? Check our FAQ on our website

<https://paiement.systempay.fr/doc/en-EN/faq/sitemap.html>

For any technical inquiries or if you need any help, contact [technical support](#).

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

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