Centre Monétique Interbancaire

Online payment integration

Version 1.4.4

Version History

Version N°	Version Date	Type of change
V1.0	05 November 2015	Creation
V1.1	19 November 2016	« encoding » parameter with « utf-8 » value
V1.2	02 February 2017	« shopurl » parameter
V1.3	02 March 2017	« currenciesList », « amountCur », « symbolCur » parameters
V1.3.1	13 April 2017	BUSINESSDATE parameter is no more used
V1.3.2	18 August 2017	ñautoRedirectò parameter
V.1.4	06 August 2018	Errors codes list + Transactions statuses + Callbacks control and monitoring
V1.4.1	12 December 2018	ñAutoRedirectò parameter name update
V1.4.2	14 February 2019	• "BillToCountry" parameter format • "BillToName" and "email" parameters are mandatory • New "sessiontimeout" parameter • Hash calculation in the case of items containing the word "document" • ISO8583-13 error code added to errors list
V1.4.3	13 May 2019	New merchantôs response ñFAILUREò to CMI callback
V1.4.4	14 January 2020	• ñdescriptionò parameterôs length updated. • Errors codes list updated. • Notice in regards to hash calculation.

Summary

1. INTRODUCTION

This document gives a detail description of the technical specifications to connect a merchant web site to the online payment gateway of the Centre Monétique Interbancaire (CMI).

2. SECURITY OF EXCHANGES

For securing the online payment transactions, CMI uses SSL protocol to encrypt the merchantôs order information, customerôs payment information and merchantôs screens in the payment platform used for transactions monitoring. For this, CMI uses SSL certificates signed by trusted certificate authorities.

Security is assured at several levels.

- On the merchantôs web site, every payment request is sent with a hash code which is generated with a hash algorithm and a secret key. This key is generated by the merchant and configured in its administrative interface of the payment platform.
- On the platform, the same hash algorithm is used with the merchantôs shared secret key to authenticate the merchantôs web site and to verify that payment requestôs elements have not been altered during their transfer across the network.
- Filling the information required for the financial processing is secured by SSL protocol with other algorithms which are not described in this document.

Cryptographic functions are not accessible to the user and thus cannot be amended or modified.

3. ON LINE PAYMENT PROCESS BY CREDIT CARD

3.1. Steps :

The process of online payment by credit card takes place according to the following steps:

- The client generates its order in the merchant site and chooses to pay online by credit card.
- The customer is redirected to the CMI platform that displays the payment screen. On this page is displayed a summary of the customerôs order with the transactionôs amount. The customer is thus invited to fill its payment credentials (payment cardôs number and expiration date and CVS code).
- When the client validates his payment form, the payment platform checks if the payment card used is authenticable. If it is the case, the customer is redirected to an authentication screen where he must fill his authentication secret code that is verified by its bank.
- In the case of a successful authentication, the payment platform sends an authorization request to the acquirer (CMI) that treats it with the issuer.
- If payment authorization is rejected (insufficient funds, the authorized cardôs ceiling exceeded, blacklisted card,é), an error message is displayed on the customerôs screen. The page allows the client to retry payment.

Notice: *The error message displayed to the client does not necessarly give the exact reason for the rejection. This is conform to the standards of fraud prevention. If the customer needs to know the reason for the payment failure, he is supposed to contact his bank. The merchant has access to the error message via its CMI back office of online transactions monitoring.*

- If the payment authorization is granted by the customerôs Bank
  - If the merchantôs web site has asked it in the initial payment request, a confirmation request for payment is sent automatically in the background (server-to-server) from CMI platform to the merchantôs web site.
  - When receiving the payment confirmationôs request, the merchant site identifies the concerned order in its database, updates its status and register all the payment information contained in the request. It also returns an acknowledgement response to the CMI platform. During this exchange, the merchantôs web site can ask the CMI platform to confirm the transaction in order to debit the customer or to cancel the transaction in order to unlock the transactionôs amount in the customerôs account.
  - In the CMI platform, an online receipt is displayed to the customer containing the order and the payment information (Order ID, Payment ID, payment date, payment method ...). This receipt can include a link to allow the client go back to the merchantôs website to finish the purchasing process.

Notice: *When a bank accept a payment authorization, it aplies a lock of the transactionôs amount on the customerôs account. It is a payment guaranty that can take 7 days. The amount is unlocked in the customerôs account in the folowing cases:*

- *When the transaction is confirmed (and the account is debited).*
- *When the merchant cancels the transaction on the same day of its authorization.*
- *If no settlement of the transaction is sent to the bank within the payment guaranty period (7 days)*

.

- In the case of confirmed transaction (manually via the merchant back office of the platform or automatically via the server-to-server payment confirmation request or via API), the transactionôs settlement is sent by the payment platform to the acquirer to debit the customerôs account and credit the merchantôs account.

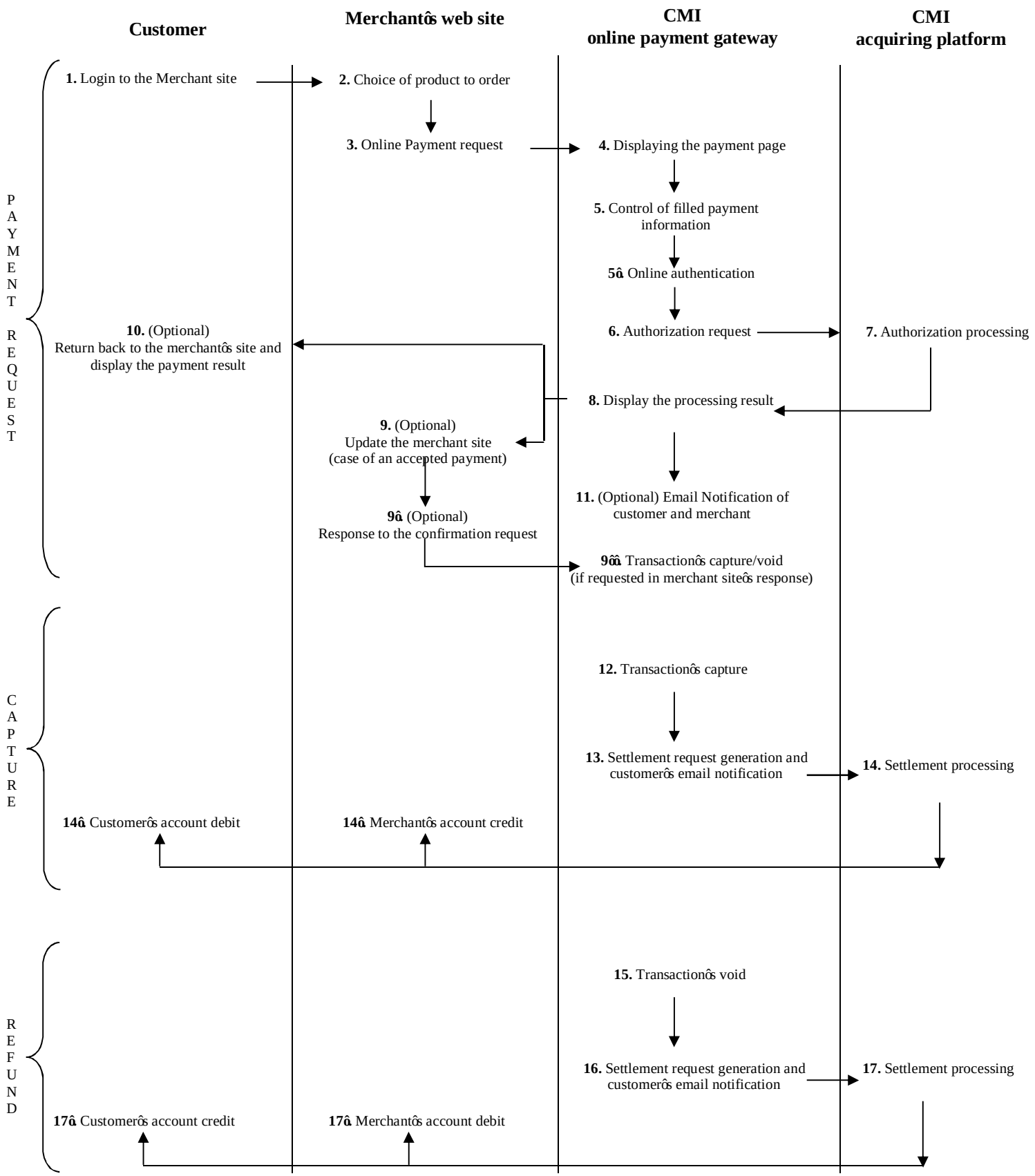
Notice: *The merchant's account is credited with the total amount of all the transactions settled by the CMI during the day. CMIôs treatment fees are deducted at source from the total amount.*

- The merchant refund the amount of a transaction to the customer via the back office of the payment platform or via API.

- Notifications are sent by the CMI platform via emails to both customers and merchants.

Notice: *The CMI platform notifications are enabled by default, but they can be disabled if the merchant wants to .*

3.2. Flow Diagram:



4. EXCHANGED DATA

The data exchanged between the merchantôs web site and the CMIôs platform is sent through an HTTP form (form post). This exchange is secured using a hash algorithm and a secret key.

4.1. PAYMENT REQUEST:

Payment request form sent from the merchantôs web site to the CMIôs platform when redirecting the customer for the online payment.

4.1.1.Parameters :

This section provides a description of the necessary attributes for the creation of the payment request. These attributes are generally organized into the following categories:

- Mandatory attributes
- Optionel attributes

Mandatory attributes

Attribute Name	Description	Format	Constraint
clientid	Merchantôs id assigned by CMI	Alphanumeric (15)	Required
storetype	Merchant payment model	Alphanumérique (15)	Required Value to use for Hosting Payment Page : « 3d_pay_hosting »
trantype	Transaction type	Alphanumérique (15)	Required Set to ñPreAuthò for preauthorization
amount	Ttransaction amount	Numeric Amount value without currency symbol. Use "." or "," as decimal separator. Example: 29.95	Required
currency	ISO code of the transaction currency	Numeric(3)	Required ISO 4217 numeric currency code. ISO code for MAD: 504
oid	Unique Order/Cart identifier in the merchant siteôs database	Alphanumeric (64)	Required

okUrl	The URL used to redirect the customer back to the merchantôs web site in case of accepted payment authorization.	URL	Required
failUrl	The URL used to redirect the customer back to the merchantôs web site in case of failed/rejected payment authorization.	URL	Required
lang	The language used to display the payment screens.	Alphabetic (2)	Required Possible values : • ar: Arabic • fr: French • en: English Default « fr »
email	Customerôs e-mail	Alphanumeric (64)	Required Email Format
BillToName	Customerôs name (first name and last name)	Alphabetic (255)	Required
rnd	Random string, will be used for hash comparison	Alphabetic (20)	Required
hash	Control hash code	Alphanumeric	Required *(See below how to generate the hash code)*
hashAlgorithm	Hash version	Alphanumeric	Required Value to use: ñver3ò

Optional attributes

Attribute Name	Description	Format	Constraint
encoding	Encoding of the posted data	Alphanumeric (32)	Optional Possible value : utf-8
description	Description sent to MPI	Alphanumeric (125)	Optional
tel	Customerôs phone number	Alphanumeric (32)	Optional
BillToStreet1	Customerôs postal adresse 1	Alphanumeric (255)	Optional
BillToStreet2	Customerôs postal adresse 2	Alphanumeric (255)	Optional
BillToCity	Customerôs city	Alphanumeric (64)	Optional
BillToStateProv	Customerôs state or province	Alphanumeric (32)	Optional

BillToPostalCode	Customerôs postal code	Alphanumeric (32)	Optional
BillToCountry	Customerôs country	Alphanumeric (3)	Optional
idN	Id of item #N, required for item #N	Alphanumeric (128)	Optional When thereôs one item, N is set to 1. In this case, the parameter name is : id1.
itemnumberN	Item number of item #N	Alphanumeric (128)	Optional
productcodeN	Product code of item #N	Alphanumeric (64)	Optional
qtyN	Quantity of item #N	Numeric	Optional
descN	Description of item #N	Alphanumeric (128)	Optional
priceN	Price of item #N	Numeric Amount value without currency symbol. Use "." or "," as decimal separator. Example: 29.95	Optional
totalN	Subtotal of item #N	Numeric Amount value without currency symbol. Use "." or "," as decimal separator. Example: 29.95	Optional
CallbackResponse	Enables/Disables the host-to-host callback request	Boolean	Optional Possible values : • true: Enabled • false: Disabled Default « true »
CallbackURL	The URL used for the host-to-host callback request	URL	Optional
shopurl	The return URL to which customer is redirected when he clicks on the button ñCancelò displayed in the payment page.	URL	Optional
currenciesList	Used to show (or not) the exchange currenciesôlist in the payment pages. This list is generated by the CMI ECOM platform. In this case, the merchant doesnôt manage the values listed. If the merchant wishs to show, in the payment page, an exchange value calculated by its web site, he should use the parameters ñamountCurò and ñsymbolCurò	Boolean	Optional Possible values : • true: Enabled • false: Disabled Default « false »

amountCur	Amount exchange (in a foreign currency) to show to the customer in the payment page. In this case, the value of the exchange amount is calculated by the merchant web site (unlike the case when the ñcurrenciesListò parameter is used) .	Numeric Amount value without currency symbol. Use "." or "," as decimal separator. Example: 29.95	Optional
symbolCur	Currency symbol to be displayed in the payment page with the ñamountCurò parameter	Alphanumeric	Optional Examples : EUR, USD.
AutoRedirect	Used in order to redirect the customer automaticaly back to the merchantôs web site when the transaction is accepted.	Boolean	Optional Possible values : • true: Enabled • false: Disabled Default « false »
sessiontimeout	Allow to set a session time-out duration for the payment page.	Numeric The parameter is calculated in seconds.	Optional Example : 600 (for a session of 10 minutes) Default: 1800 seconds. Minimum allowed value is 30 seconds and the maximum is 2700 seconds.

Notices:

- *It is recommended that the orderô status update in the merchantô web site uses the server-to-server payment confirmation request via the CallbackURL parameter. The parameters okUrl and failUrl can also be used for this purpose, but they require customers to be redirected back (html) to the merchantô web site.*
- *The URL set in the parameter CallbackURL must be reachable on Internet even in testing phase.*
- *The URL set in the CallbackURL parameter should not require authentication (login/password).*
- *It is strongly recommended to use the "encoding" parameter, even if it is optional, to avoid problems when the data contains special characters.*
- *Care must be taken to respect the size of each parameter of the payment request.*

4.1.2.Payment request Example

```
<form method="post" action="https://host/fim/est3dgate ">
    <input type="hidden" name="clientid" value="990000000000001"/>
    <input type="hidden" name="storetype" value="3d_pay_hosting" />
```

```
    <input type="hidden" name="hash" value="iej6cPOjDd4IKqXWQEznXWqLzLI=" />
    <input type="hidden" name="trantype" value="PreAuth" />
    <input type="hidden" name="amount" value="31.50" />
    <input type="hidden" name="currency" value="504" />
    <input type="hidden" name="oid" value="1291899411421" />
    <input type="hidden" name="okUrl" value="https://www.teststore.ma/success.php" />
    <input type="hidden" name="failUrl" value="https://www.teststore.ma/fail.php" />
    <input type="hidden" name="lang" value="en" />
    <input type="hidden" name="rnd" value="asdf" />
    <input type="hidden" name="hashAlgorithm" value="ver3">

Optional parameters
    <input type="hidden" name="tel" value="012345678">
    <input type="hidden" name="email" value="test@test.ma">

<!-- Billing Parameters [All Optional]-->

        <input type="hidden" name="BillToCompany" value="Billing Company">
        <input type="hidden" name="BillToName" value="Bill John Doe">
        <input type="hidden" name="BillToStreet1" value="Address line 1">
        <input type="hidden" name="BillToStreet2" value="Address line 2">
        <input type="hidden" name="BillToCity" value="Casablanca">
        <input type="hidden" name="BillToStateProv" value=" Casablanca">
        <input type="hidden" name="BillToPostalCode" value="12345">
        <input type="hidden" name="BillToCountry" value="504">

<!-- Order Item Parameters [All Optional]-->
       <input type="hidden" name="ItemNumber1" value="a5">
       <input type="hidden" name="ProductCode1" value="a5">
       <input type="hidden" name="Qty1" value="3">
       <input type="hidden" name="Desc1" value="a5 desc">
       <input type="hidden" name="Id1" value="a5">
       <input type="hidden" name="Price1" value="10.50">
      <input type="hidden" name="Total1" value="31.50">

</form>
```

4.1.3.Hash generation of the payment request

Hashing Approach is used in order to authenticate users for transaction requests. To generate the hash for client authentication, append all posted request parameter values in alphabetical order (A to Z) using pipeline ñò as the separator between parameters. If a parameter is sent to CMI platform with an empty value, it still will be added to hash data (For an empty value example, email parameter value can be checked in below example).

After all request parameters are added alphabetically using pipeline ñò as separator, storeKey will be added to hash data again using pipeline ñò

Note that in a case of using ñò character in a parameter, ñò character is used for escaping. Therefore, if the ñò character is also used in a parameter as its own value, we use ñò character before it and then append it to hash plain text. For better understanding, you can check the example below :

Original Value	: ORDER-256712jbs\j6b|
Value used for Hash Calculation	: ORDER-256712jbs\\j6b\\\|

Example Parameters and Hash Calculation:

clientId	100200127
amount	95.93
okurl	http://localhost:8080/SampleCodeJSPTest/GenericVer3ResponseHandler
failUrl	http://localhost:8080/SampleCodeJSPTest/GenericVer3ResponseHandler
TranType	PreAuth
email	
callbackUrl	http://localhost:8080/SampleCodeJSPTest/GateResponseControl.jsp
currency	504
rnd	87954458746
storeType	3d_pay_hosting
lang	en
hashAlgorithm	ver3
BillToName	name
BillTocompany	billToCompany
storeKey	ABCD1234

Hash:

Order of Used Parameters in Hash Data :

amount|BillToCompany|BillToName|callbackUrl|clientid|currency|email|failUrl|hashAlgorithm|lang|okurl|rnd|storetype|TranType|storeKey

Plaintext:

95.93|billToCompany|name|http://localhost:8080/SampleCodeJSPTest/GateResponseControl.jsp|100200127|504||http://localhost:8080/SampleCodeJSPTest/GenericVer3ResponseHandler|ver3|en|http://localhost:8080/SampleCodeJSPTest/GenericVer3ResponseHandler|87954458746|3d_pay_hosting|PreAuth|ABCD1234

Hash = Base64(SHA512(plaintext))

Notice:

- *Parameters named ñencodingò and ñhashò are ignored during hash calculation.*
- *For security reasons, the CMI platform uses a tool that detects, prevents and disables the execution of unauthorized scripts (code contained in the HTTP parameters). So you have to replace any character after the string "document" with a dot "." during the calculation of the hash.*
  Examples:
  document abc -> document.abc
  documentabc -> document.bc
- *In order to avoid a hash error, you must ensure that the values of the calculated parameters are the same as those sent in payment request.*
  For example, adding a submit button that will perform the action of sending the payment form may influence the hash calculation. To remedy this case, you must calculate the value of this element (button) in the hash calculation or delete its attribute "name".

4.2. PAYMENT RESULT :

When the customer submits the payment page with his card credentials, the 3D authentication flow starts. The payment authorization is then been processed with the customerôs issuer bank. Payment response parameters and some or all parameters that were sent by the merchantôs web site will be posted back to the merchantôs web site by the CMI platform. In case of an accepted payment, the merchantôs web site updates the order status in its DB and can ask the CMI platform to confirm the payment for the customer to be debited.

4.2.1.Host-to-host callback:

The merchant starts payment flow by sending a PreAuth request to CMI platform with the parameter CallbackResponse set to the value ñtrueò (CallbackResponse=true). In addition to the required parameters, CallbackUrl should be sent so that CMI platform can send the confirmation callback to the merchant.

What should be noticed for this callback request in server-to-server mode, is that it reminds the merchantôs web site as well of the cases of rejections as cases of successful transactions. So it may well be that the merchantôs web site receives, for the same transaction (same order number), rejection returns that will be followed by a return of transaction acceptance. This means that the client has failed attempts before it succeeds. It is the parameter ProcReturnCode which makes it possible to distinguish between callback requests of successful payment authorizations (ProcReturnCode = 00) and the callback requests of the payment failures (ProcReturnCode! = 00 or nonexistent).

Merchantôs web site response:

Upon receiving the CMI callback request, the merchantôs web site is expected to send callback response to CMI platform. The following callback responses are supported:

- **Acknowledgement:** Merchantôs response should be ñAPPROVEDò
  The transaction is acknowledged by the merchant but he asks to not debit the customer. The merchant needs to manage capture or void manually via CMI Merchant Center interface.

- **Acknowledgement&Settle:** Merchantôs response should be ñACTION=POSTAUTHò
  In this case, an automatic capture (PostAuth) request will be sent to the acquirer bank for the customer to be debited.

- **Failure:** Merchantôs response should be ñFAILUREò
  The merchant website fails to take into account the transaction. The customer is asked to check the status of his/her order on the merchantôs site. The merchant needs to manage capture or void manually via CMI Merchant Center interface.

- **Timeout:** If timeout occurs with the merchantôs web site, the customer is asked to check the status of his/her order on the merchantôs web site. Merchant needs to manage capture or void manually via Merchant Center interface.

- **Merchantôs response syntax error:** If there is an error from the merchantôs web site, the customer is asked to check the status of his/her order in the merchantôs web site. Merchant needs to manage capture or void manually via Merchant Center interface.

Merchantôs web site controls:

During the callback request, the merchantôs web site is supposed to do the following:

- Generate a hash code with the same data posted by the CMI platform in the callback request. Then compare this calculated hash with the hash sent by the CMI platform in the callback request. If they are identical, proceed to the next check.

- Look, in the ordersôDB of the merchantôs web site, for the record identified by the value of the "oid" parameter sent by the CMI platform in the callback request.

- Check if the amount of the order recorded in the ordersôDB of the merchantôs web site is equal to the amount sent by the CMI in the callback request via the "amount" parameter.

- Check the "ProcReturnCode" parameter value sent by the CMI in the callback request:
  - If ProcReturnCode = 00, this is an accepted transaction.

- So it is necessary to update the status of the order in the ordersôDB of the merchantôs web site (status = Paid).
- Answer the CMI callback request with:
  - "ACTION=POSTAUTH": in order to debit the client automatically.
  - "APPROVED": in order to not debit the client automatically. In this case, the merchant needs to manage capture or void manually via CMI Merchant Center interface.

- If the ProcReturnCode <> 00 or if ProcReturnCode parameter does not exist in the callback request, it is a payment authorization failure.
  - In this case, do not change the status of the order in the BDD orders of the merchantôs web site.
  - The response to return to the CMI callback request is "APPROVED" (acknowledgment).

- If a technical problem occurs in one of the previous steps, answer the CMI callback request with "FAILURE". In this case, the merchant needs to manage capture or void manually via CMI Merchant Center interface.

Notice:

- *The callback request sent by CMI platform to the merchantôs web site in server-to-server mode, reminds as well of the cases of successful transactions as cases of rejections. So it may well be that the merchantôs web site receives, for the same transaction (same order number), rejection returns that will be followed by a return of transaction acceptance. This means that the client has failed attempts before it succeeds. It is the parameter ProcReturnCode which makes possible to distinguish between the callback request of a successful payment authorization (ProcReturnCode = 00) and the callback request of a failed payment (ProcReturnCode! = 00 or nonexistent).*

4.2.2.Redirection back to merchant web site:

When a transaction is processed, a link is shown in the payment result page so that the customer can go back to the merchantôs web site.

In the case of an approved transaction, the customer will be redirected to **okUrl** (parameter sent by the merchant in the payment request). All input parameters along with transaction response parameters will be posted to **okUrl**, and the ñResponseò parameter (see below) value will be ñ**Approved**ò

In the case of a failed transaction, the customer will be redirected to **failUrl** (parameter sent by the merchant in the payment request). All input parameters along with transaction response parameters will be posted to **failUrl**, and the ñResponseò parameter (see below) will be ò**Declined**ò or ñ**Error**ò

Notice:

- *In general, the update of the payment status of the order at the merchant website is done at the time of the server-to-server callback request (via CallbackURL). It is recommended that the merchant website also checks the payment status of the order when the customer is redirected to okURL or failURL. Indeed, in the case where the request of the callback server-to-server (via the CallbackURL) fails, the redirection of the customer to the okURL or the failURL can be used to take into account the payment status that was processed in the CMI platform. We have just to remember that in the case of the server-to-server callback request (via the CallbackURL), the CMI platform expects a response from the merchant website to decide whether the customer should be debited or not; while in the case of customer redirection to okURL or failURL, no response is expected from the merchant website. So in this case the merchant must manually process the confirmation of the transaction via its Merchant Center CMI to debit the customer if necessary.*

4.2.3.Payment result requestôs parameters:

When CMI platform sends back the payment result to the merchant web site (either via host-to-host callback or via customer redirection back), the request contains, in addition to the parameters sent by the merchant in the payment request, the following parameters:

Attribute name	Description	Format	Constraint
Response	Payment status	Alphabetic	Possible values: "Approved", "Error", "Declined"
ProcReturnCode	Transaction status code	Alphanumeric (2)	Possible values: ñ00ò for authorized transactions, ñ99ò for gateway errors, others for ISO-8583 error codes
EXTRA.TRXDATE	Transaction Date	Alphanumeric (17)	Formatted as "dd/mm/0yyyy hh24:mi:ss"
AuthCode	Transaction Verification/Approval/Authorization code	Alphanumeric (6)	
acqStan	A payment identifier (Payment ID) shared with acquirer and issuer	Numeric (6)	
HostRefNum	Host reference number	Alphanumeric (12)	
TransId	CMI platform Transaction Id	Alphanumeric (64)	
ErrMsg	Error message	Alphabetic (255)	
ClientIp	IP address of the customer	Alphanumeric (15)	Formatted as "###.###.###.###"
ReturnOid	Returned order ID	Alphanumeric (64)	Must be the same as input oid (sent in the payment request of the merchant)
paymentType	Used payment method	Alphanumeric	Possible values: CARD
EXTRA.CARDBRAND	Card brand name	Alphabetic	Possible values: VISA, MASTERCARD
MaskedPan	Masked credit card number	Alphanumeric (19)	Formatted as XXXXXX******XXXX
cardHolderName	Card holden name used by the customer when processing the transaction	Alphanumeric	
Ecom_Payment_Card_ExpDate_Year	Card expiry year	Numeric (2)	
Ecom_Payment_Card_ExpDate_Month	Card expiry month	Numeric (2)	
EXTRA.CARDISSUER	Card issuer name	Alphabetic	
merchantID	MPI merchant ID	Alphanumeric (15)	
ACQBIN	Acquirer BIN	Numeric (6)	

mdStatus	Status code for the 3D transaction	Numeric (1)	Possible values: 1=authenticated transaction (Full 3D) 2, 3, 4 = Card not participating or attempt (Half 3D) 5,6,7,8 = Authentication not available or system error 0 = Authentication failed
txstatus	3D status for archival	Alphabetic (1)	Possible values "A", "N", "Y"
iReqCode	Code provided by ACS indicating data that is formatted correctly, but which invalidates the request. This element is included when business processing cannot be performed for some reason.	Numeric (2)	
iReqDetail	May identify the specific data elements that caused the Invalid Request Code (so never supplied if Invalid Request Code is omitted).	Alphanumeric	
vendorCode	Error message describing iReqDetail error.	Alphanumeric	
PAResSyntaxOK	If PARes validation is syntactically correct, the value is true. Otherwise value is false.	Alphabetic (1)	Possible values "N", "Y"
ParesVerified	If signature validation of the return message is successful, the value is true. If PARes message is not received or signature validation fails, the value is false.	Alphabetic (1)	Possible values "N", "Y"
eci	Electronic Commerce Indicator	Numeric (2)	Empty for non-3D transactions
cavv	Cardholder Authentication Verification Value, determined by ACS.	Alphanumeric (28)	Contains a 20 byte value that has been Base64 encoded, giving a 28 byte result.
xid	Unique internet transaction ID	Alphanumeric (28)	Base64 encoded
cavvAlgorthm	CAVV algorithm	Numeric (1)	Possible values "0", "1", "2", "3"
md	MPI data replacing card number	Alphanumeric	
Version	MPI version information	Alphanumeric (3)	Like "2.0"
sID	Schema ID	Numeric (1)	Possible values: "1" for Visa, "2" for Mastercard
MdErrorMsg	Error Message from MPI (if any)	Alphanumeric (512)	
rnd	Random string, will be used for hash comparison	Alphanumeric (20)	
HASH	Hash value	Alphanumeric (20)	

4.2.4.Hash generation of the payment result

(Use the same technique of the payment request hashôs generation ï See above)

4.2.5.Callback requests data examples

Successful transaction case:

Parameter	Value	Parameter	Value
ACQBIN	439218	INVOICENUMBER	
acqStan	56928	iReqCode	
amount	27.47	iReqDetail	
AuthCode	746579	itemnumber1	a1
Bill2Email	test@cmi.co.ma	itemnumber2	b1
BillToCity	Casablanca	itemnumber3	c1
BillToCompany	Billing Company	lang	en
BillToCountry	504	MaskedPan	400000***7190
BillToName	Bill John Doe	md	400000:BD64FA1BA77717 D21DA26554D2DE0 C9E65AB5D5872737AE720 F5D2484180228C: 4024:##600000004
BillToPostalCode	12345	mdErrorMsg	
BillToStateProv	mystate	mdStatus	1
BillToStreet1	Address line 1	merchantID	600000004
BillToStreet2	Address line 2	MERCHANTSURCHARGE	
BillToStreet3	Address line 3	noCallbackPaymentMethodList	
BillToTelVoice	123456	oid	sfgzzy4
CallbackResponse		okUrl	http://test.cmi.co.ma/ok
callbackUrl	http://test.cmi.co.ma/callback	PAResSyntaxOK	
cardHolderName	Cardholder name	PAResVerified	
cavv	AAABCRA0l1WFQDgnWDSX AAAAAAA=	paymentType	CARD
cavvAlgorithm		payResults.dsId	1
choix1	on	pMethod	
clientid	600000004	price1	2.00
clientIp	81.192.141.16	price2	3.95
currency	504	price3	3.50
CUSTOMERMSISDN		ProcReturnCode	0
CUSTOMERSURCHARGE		productcode1	a2
CVVPresence	1	productcode2	b2
desc1	desc1	productcode3	c2
desc2	desc2	qty1	3
desc3	desc3	qty2	1
digest	digest	qty3	5
DIMCRITERIA1		RecurringFrequency	
DIMCRITERIA10		RecurringFrequencyUnit	
DIMCRITERIA2		RecurringPaymentNumber	
DIMCRITERIA3		refreshtime	10
DIMCRITERIA4		Response	Approved
DIMCRITERIA5		ReturnOid	sfgzzy4

DIMCRITERIA6		rnd	lbJjfQCTTrNRfMcNe1I1
DIMCRITERIA7		sessiontimeout	
DIMCRITERIA8		SettleId	1
DIMCRITERIA9		ShipToCity	Casablanca
dsId	1	ShipToCompany	Shipping Company
eci	5	ShipToCountry	504
Ecom_Payment_Card_ExpDate_Month	12	ShipToName	Ship John Doe
Ecom_Payment_Card_ExpDate_Year	17	ShipToPostalCode	12345
EDITABLEORDERITEM		ShipToStateProv	mystate
ErrMsg		ShipToStreet1	Address line 1
EXCHANGEAMOUNT	@@EXCHANGEAMOUNT@@	ShipToStreet2	Address line 2
EXCHANGECURRENCY		ShipToStreet3	Address line 3
EXTRA.CARDBRAND	VISA	ShipToTelVoice	789456
EXTRA.CARDISSUER	CDM	shopurl	http://test.cmi.co.ma/shop
EXTRA.HOSTMSG	APPROVED	SID	
EXTRA.TRXDATE	23/11/02017 15:57:06	storetype	3D PAY HOSTING
failUrl	http://test.cmi.co.ma/fail	total1	6.00
fatouratiExpress	1	total2	3.95
GRACEPERIOD		total3	17.50
hashAlgorithm	ver3	TRANID	129430
HostRefNum	732715056928	TransId	17327P7GH13718
id1	id1	trantype	PreAuth
id2	id2	txstatus	Y
id3	id3	vendorCode	
instalment		version	
		xid	7AFKUXDa5KeeIWxM6wxfB9YfLDY=

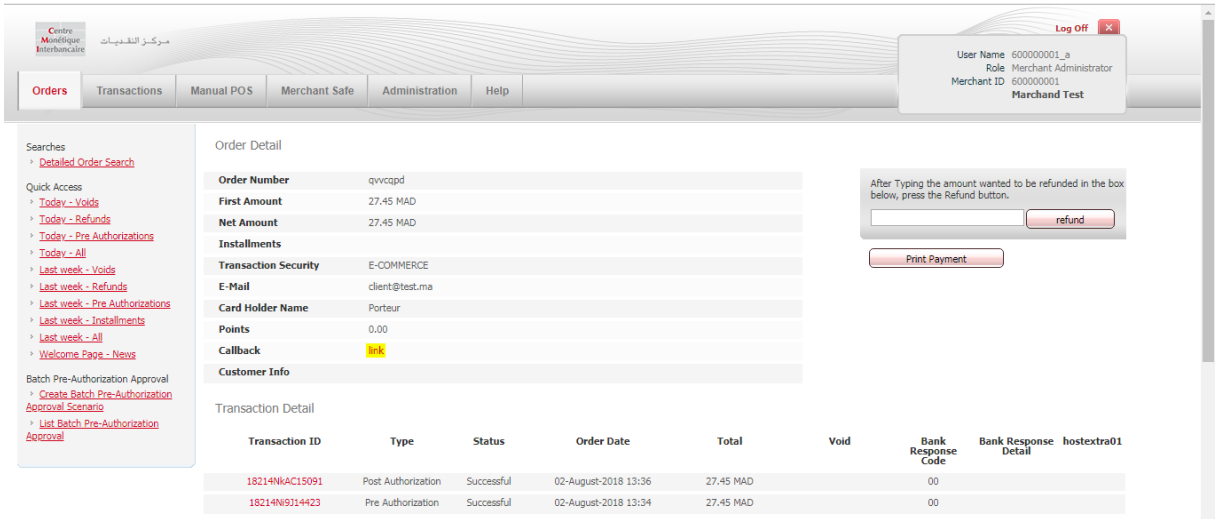
Failed transaction case:

Parameter	Value	Parameter	Value
BillToCompany	Company	mdErrorMsg	Multiple values exist for the single paramater.
BillToName	Name	mdStatus	7
callbackUrl	http://test.cmi.co.ma/callback	oid	12345
clientid	600000000	okUrl	http://test.cmi.co.ma/ok
clientIp	1.1.1.1	price	
currency	504	ProcReturnCode	99
CVVPresence	1	productCode	
Ecom_Payment_Card_ExpDate_Month	1	qty	
Ecom_Payment_Card_ExpDate_Year	20	refreshtime	300
email	test@cmi.co.ma	ResponseError	

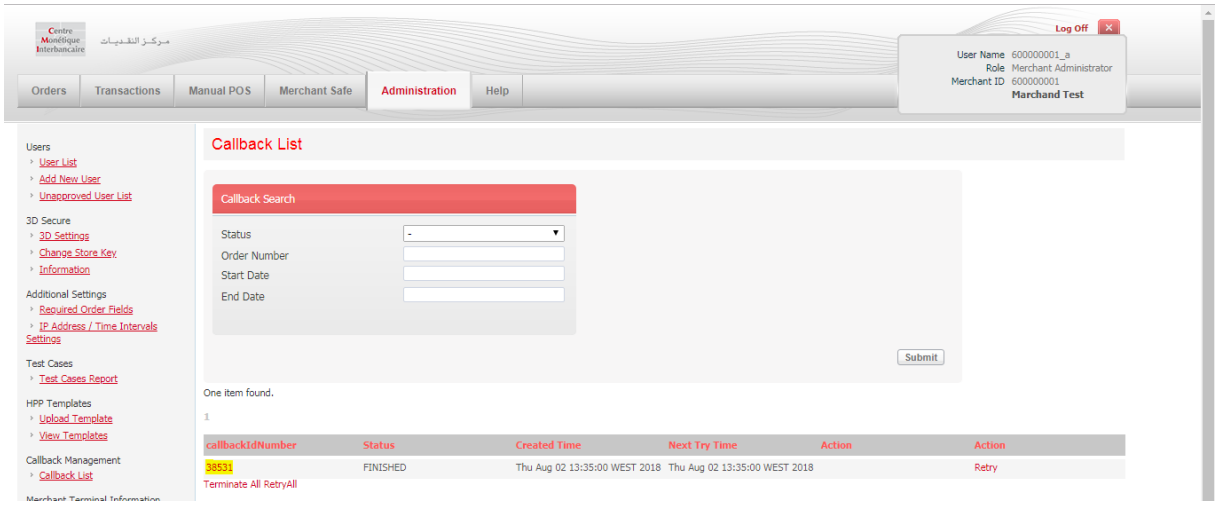
ErrCode	CORE-5110	retryXid	I6Hjx1K6zUGjqY98Z+vEnR2/gr4=
ErrMsg	Multiple values exist for the single paramater.	rnd	mmduZ3aMFe8qDmEG1MV1
failUrl	http://test.cmi.co.ma/fail	shopurl	http://test.cmi.co.ma/shop
hashAlgorithm	ver3	storetype	3d_pay_hosting
itemNumber		tel	600000000
lang	fr	total	
MaskedPan	400000***7190	trantype	PreAuth
		xid	I6Hjx1K6zUGjqY98Z+vEnR2/gr4=

4.2.6.Callbacks monitoring via CMI Merchant Center

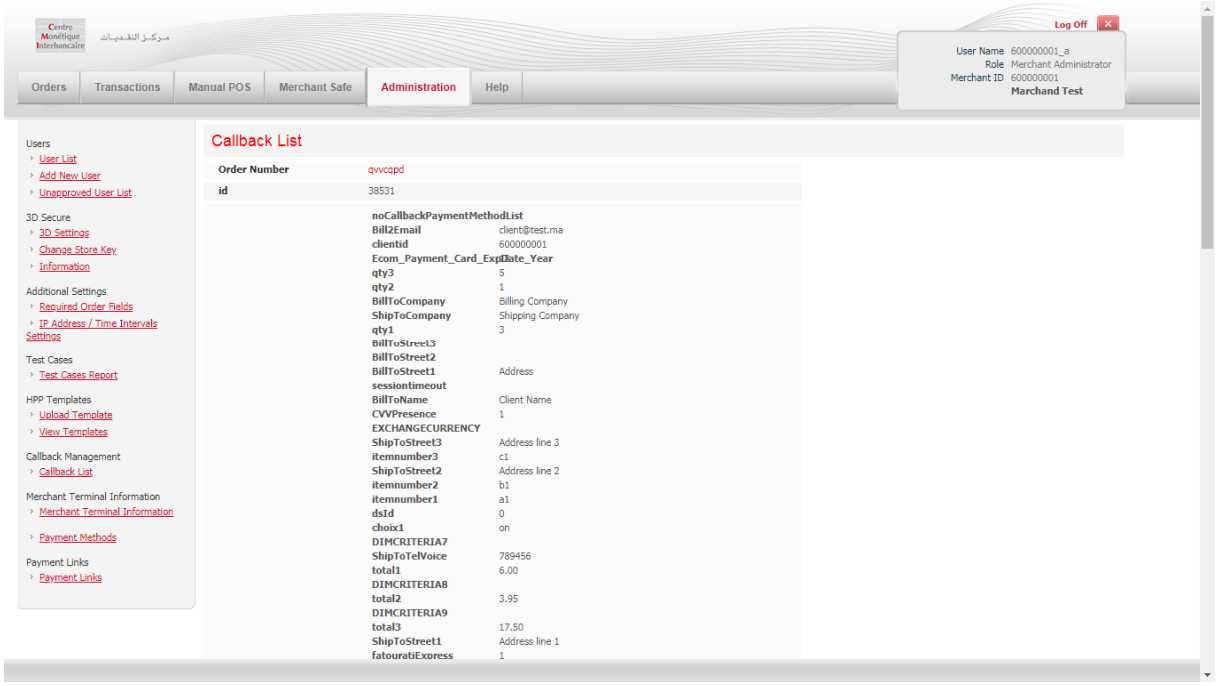
The merchant has the possibility to view the details of the callback requests through his Merchant Center CMI where he is used to track his online transactions.



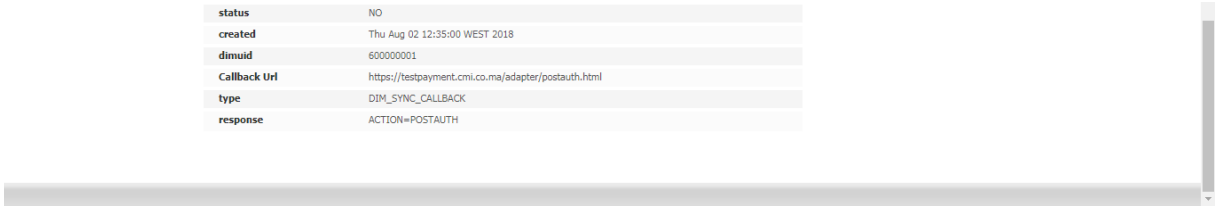
By clicking on the Callback link, in the order details, we can access the detail of the callback request that is exchanged between the CMI platform and the merchant website.



We have to click on the callback's identifier to be able to access its detail.



We can then see all the data sent by the CMI platform to the merchant website in the callback request.



We can even see the response of the merchant site to the request of the Callback of the CMI.

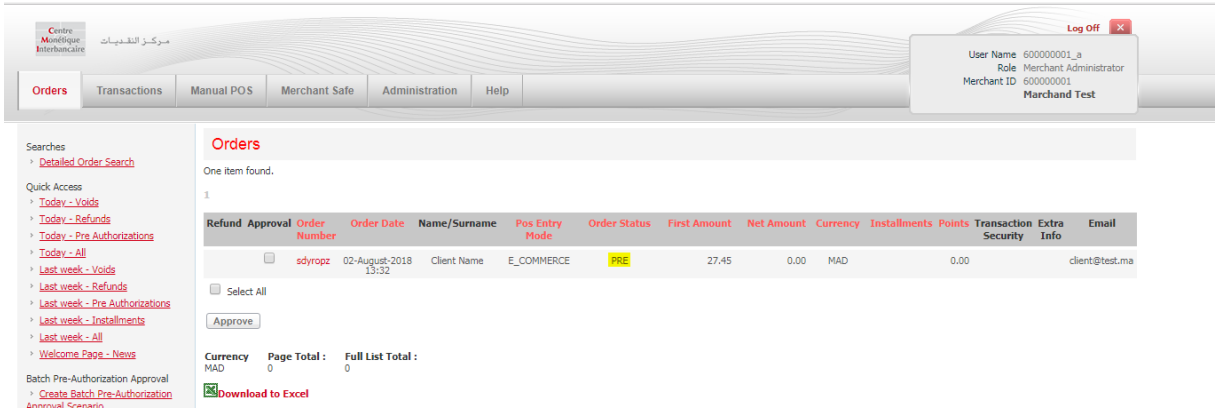
5. TRANSACTIONS' STATUSES

The payment process goes through several steps during which the transaction status changes. In this chapter, we detail this topic with illustrations from the CMI Merchant Center which is made available by CMI to merchants so that they can track their online transactions at any time.

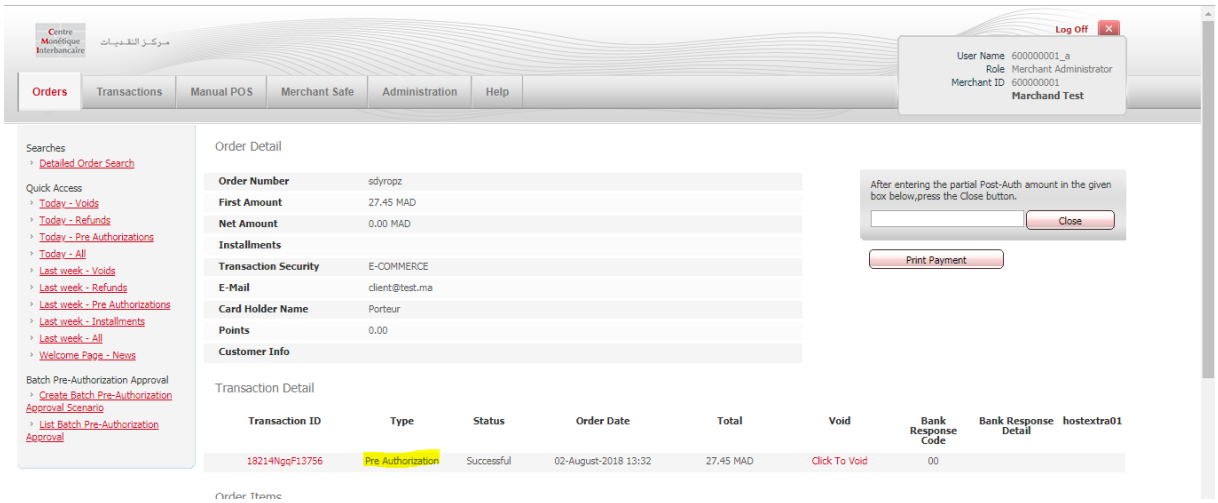
5.1. PRE AUTHORIZATION :

When the customer enters his payment card details in the payment page, the CMI sends a request for payment authorization to the issuing bank. When the bank accepts the request, it blocks the transaction's amount in the customer's bank account. In this case, the customer is not charged. It is indeed a blocking of funds which guarantees the merchant the recovery of its funds if he accepts the transaction within the allowed time (up to 4 or 7 days).

At this stage, the transaction is processed within the Pre Authorization phase. This is visible through the Merchant Center CMI as shown below.



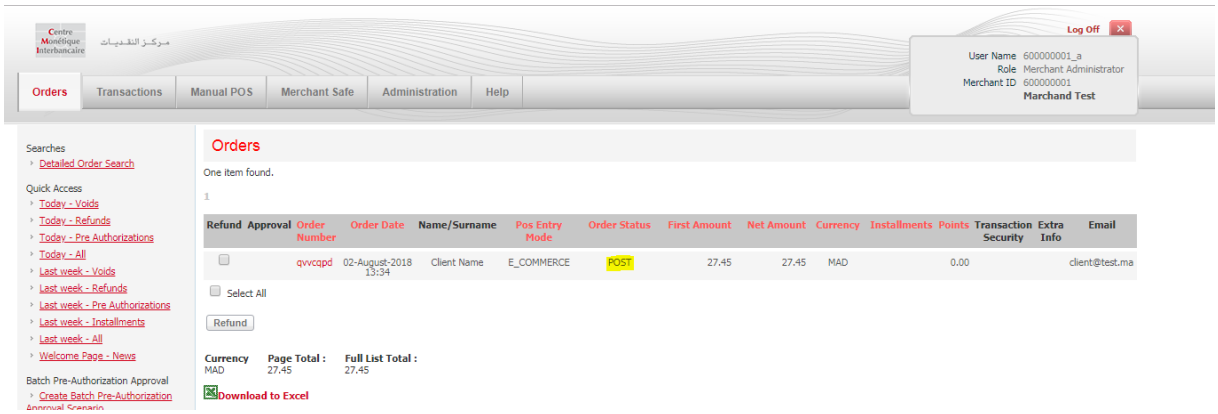
The order status is set to "PRE" which refers to the Pre Authorization.



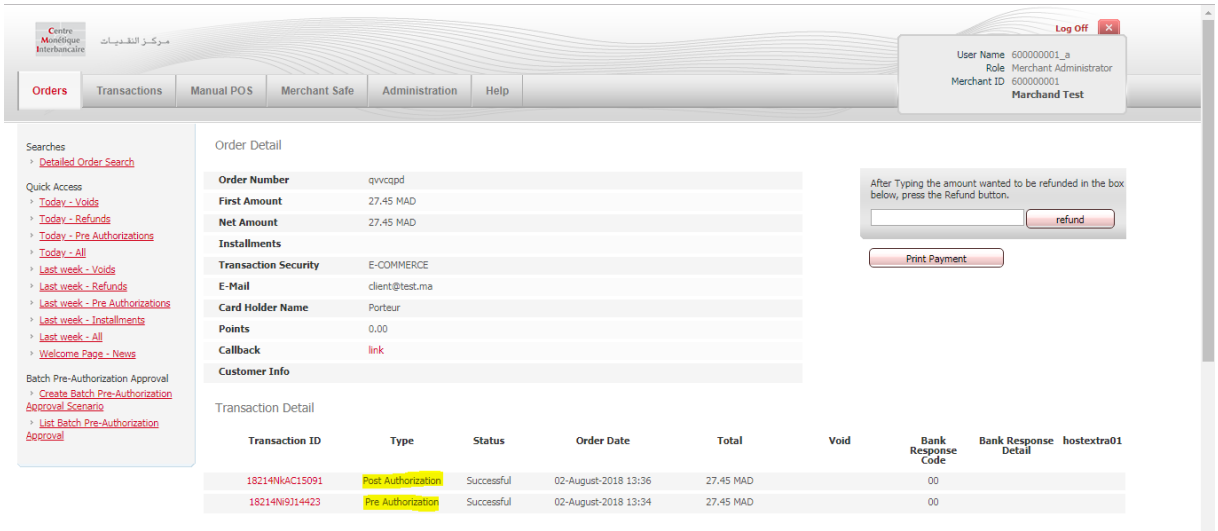
In the order details, we find a single transaction whose type is "Pre Authorization".

5.2. POST AUTORIZATION :

When the CMI platform sends a callback request to the merchant website to return the payment result, the merchant website requests to debit the customer by replying with "ACTION=POSTAUTH". In this case, the transaction moves to the Post Authorization phase. This is visible through the Merchant Center CMI as shown below.



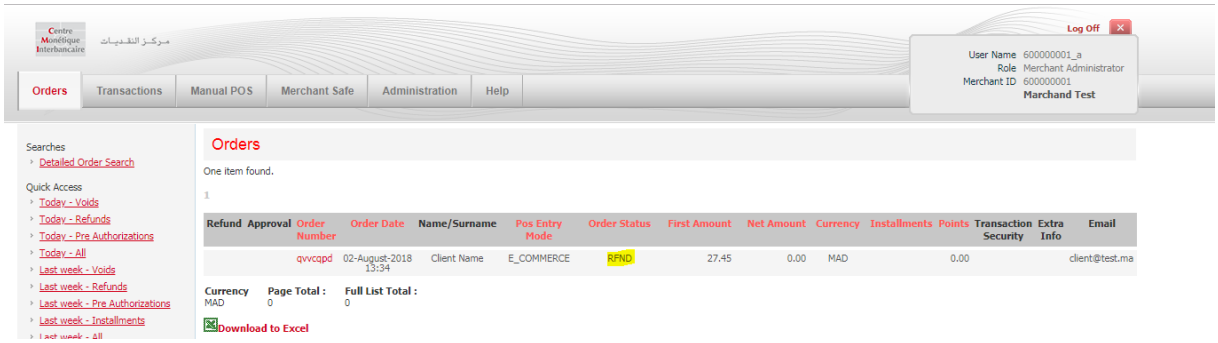
The order status is set to "POST" which refers to the Post Authorization.



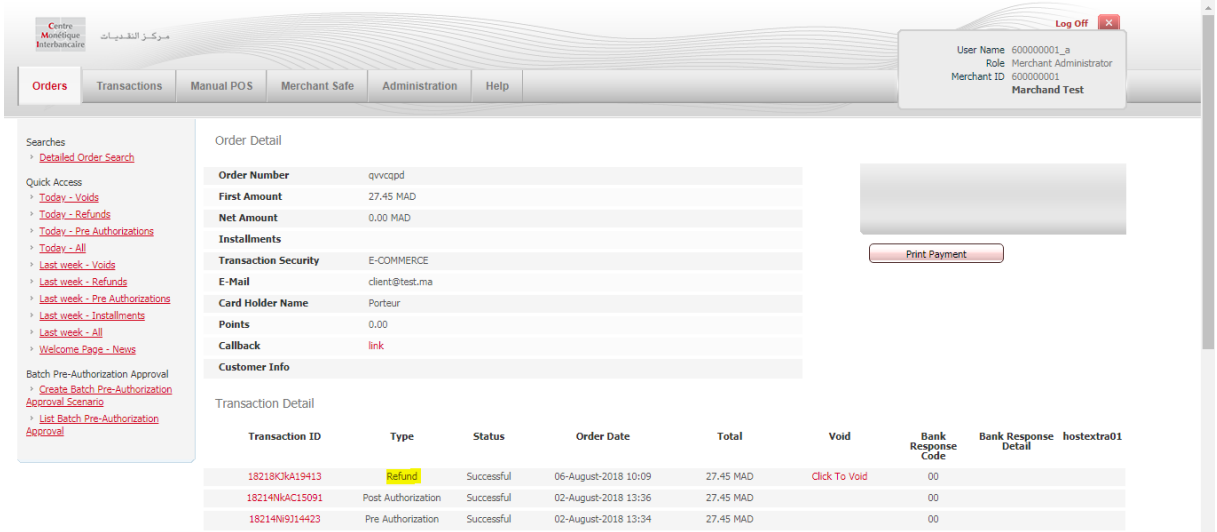
In the order detail, there are two transactions. One of them is set to type "Pre Authorization" and the other one is set to type "Post Authorization".

5.3. REFUND :

Sometimes, the merchant wants to payback one of his customers that has processed an online transaction through the merchant's website. The merchant can in this case execute a refund operation through its CMI Merchant Center. Once executed, the transaction moves to the Refund phase. This is visible through the CMI Merchant Center as shown below.



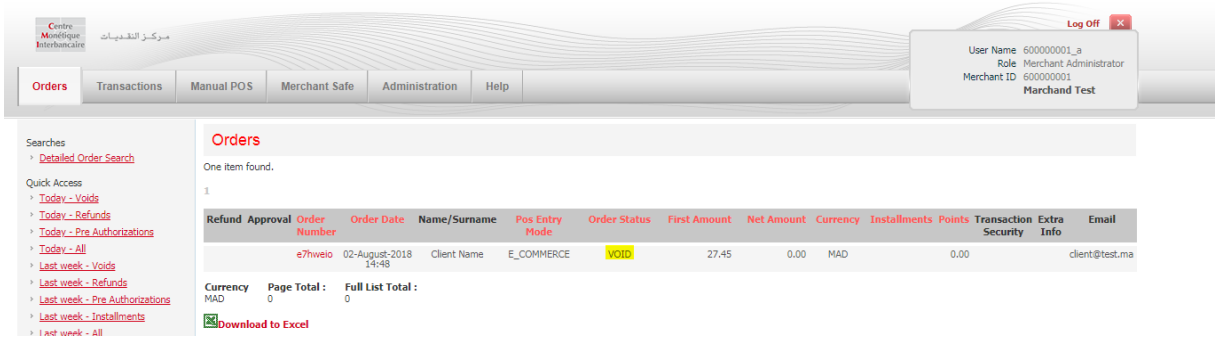
The order status is set to "RFND" which refers to the Refund.



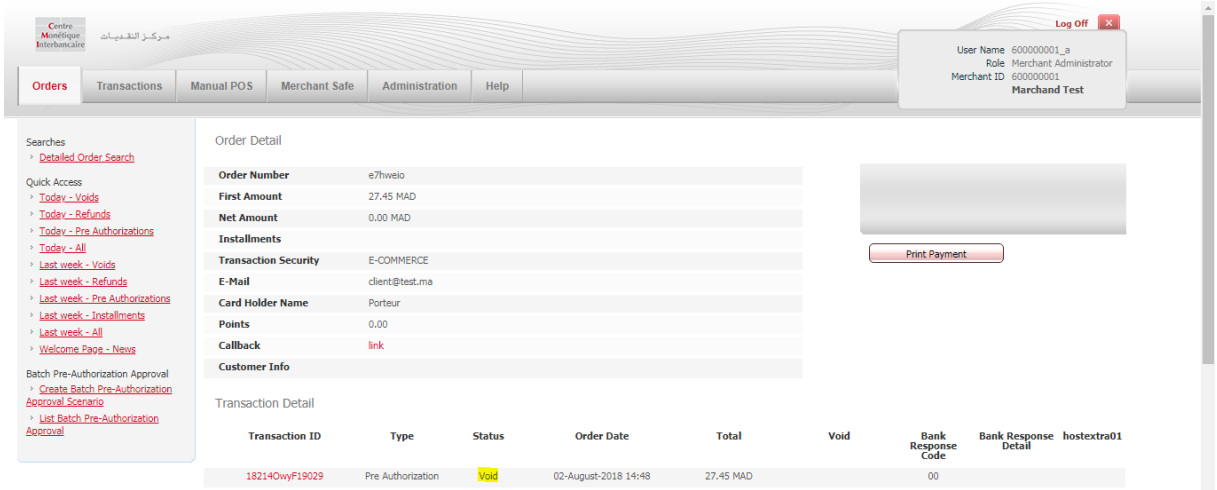
In the order detail, there are three transactions. One of them is set to type "Refund" and the other ones are set to types "Pre Authorization" and "Post Authorization".

5.4. VOID :

When the transaction is in the "Pre Authorization" phase, the merchant can execute its cancellation to request the release of funds at the customer's bank account. The cancellation operation can be done through the CMI Merchant Center. This is visible through the Merchant Center CMI as shown below.



The order status is set to "VOID" which refers to the Cancellation.



In the order detail, the transactionôs type is ñPre Authorizationòwith the status ñVoidò

Notice:

- *For more information on Merchant Center CMI features, please refer to the appropriate user manual.*

6. ANNEX

6.1. *CMI ECOM Integration serviceôs contact :*

If you have a question about the technical interfacing of your web site with CMI online payment platform, you can contact our integration service via the following email:
integration.ecom@cmi.co.ma

6.2. *Error codes :*

The following table lists the main errors codes that identify transactionsôfailurs and that can be found in the CMI Merchant Center with all other transactionsôdetails:

Error Code	System Error Message	Err Desc EN	Err Desc FR	Err Desc AR	Comments
99	3D-1004	Wrong security code	Code de sécurité erroné	[illegible]	Rejection of the platform due to hash code calculation error. This can happen either in the payment request, or in the payment result request.
99	3D-1005	Opertaion failor	Echec de l'opération	[illegible]	Rejection of the platform due to cardholder online authentication failure.
99	3D-1034	Opertaion failor	Echec de l'opération	[illegible]	Rejection of the platform due to mishandling of the client.
99	BM-1002	HOST based messaging problem	HOST based messaging problem	HOST [illegible]	Invalid response message from acquirer.
99	BM-9101	Failed operation	Opération échouée	[illegible]	Acquiring interface timeout.
99	BM-9102	Failed operation	Opération échouée	[illegible]	Acquiring interface timeout.
99	CORE-2010	The credit card is expired.	La carte de crédit est expirée	[illegible]	Rejection of the platform due to expired credit card use.
99	CORE-2012	The credit card number is not in a valid format.	Le format du numéro de carte de crédit n'est pas valide.	[illegible]	Rejection of the platform.
99	CORE-2202	Failed operation	Opération échouée	[illegible]	Rejection of the platform due to a control rule.

					Example of control rules: Authorization request in case of 3D Secure authentication fallback.
99	CORE-2208	Card brand is not allowed.	Marque de carte non autorisée.	[illegible]	Rejection of the platform.
99	CORE-2253	The payment authorization could not be performed	Impossible de procéder à l'autorisation de paiement	[illegible]	Rejection of the platform due to a control rule. Example of control rules: Using a non-authenticatable 3D Secure foreign payment card.
99	CORE-2515	Incorrect data	Données incorrectes	[illegible]	Rejection of the platform due to bad data entered by the customer (Exp. Bad CVV Format).
99	CORE-5110	Please return to the web site and try again	Merci de retourner au site web et réessayer	[illegible]	Rejection of the platform due to mishandling of the client.
99	RULE-0001	Your session has expired. Please return to the merchant website and try again.	Votre session a expiré. Merci de revenir au site marchand et réessayer.	[illegible]	CMI platform rejection due to session timeout.
99	RULE-0002	Cardholder authentication required	Authentification du porteur requise	[illegible]	CMI platform rejection due to cardholder authentication lack.
99	RULE-0003	Wrong card type	Type de carte erroné	[illegible]	CMI platform rejection due to wrong card type use.
99	RULE-0004	Card not allowed	Carte non permise	[illegible]	CMI platform rejection due to not allowed card use.
03	ISO8583-03	Payment authorization not permitted for this merchant	Autorisation de paiement non permise pour ce marchand	[illegible]	Bank rejection.
04	ISO8583-04	Payment authorization rejected with this card	Autorisation de paiement rejetée avec cette carte	[illegible]	Rejection of the bank for stolen credit card use.
05	ISO8583-05	Payment authorization declined	Autorisation de paiement non acceptée	[illegible]	Rejection of the bank without precision of the reason.
12	ISO8583-12	Payment authorization rejected	Autorisation de paiement rejetée	[illegible]	Bank rejection.
13	ISO8583-13	INVALID AMOUNT	Montant non valide	[illegible]	Bank rejection.

14	ISO8583-14	Payment authorization rejected with the used card	Autorisation de paiement rejetée avec la carte utilisée	[illegible]	Bank rejection.
15	ISO8583-15	Payment authorization failed	Autorisation de paiement échouée	[illegible]	Bank rejection.
39	ISO8583-39	No credit account	Pas de compte de crédit	[illegible]	Bank rejection.
51	ISO8583-51	Insufficient funds.	Solde de la carte insuffisant	[illegible]	Bank rejection.
54	ISO8583-54	Payment authorization rejected with this card	Autorisation de paiement rejetée avec cette carte	[illegible]	Rejection of the bank due to expired card use or incorrect card expiry date use.
57	ISO8583-57	Payment authorization not permitted	Autorisation de paiement non permise	[illegible]	Bank rejection.
61	ISO8583-61	Activity amount limit exceeded.	Plafond dépassé	[illegible]	Bank rejection.
62	ISO8583-62	Payment authorization rejected with this card	Autorisation de paiement rejetée avec cette carte	[illegible]	Bank rejection.
63	ISO8583-63	Payment authorization rejected with this card	Autorisation de paiement rejetée avec cette carte	[illegible]	Rejection of the bank due to security reason.
65	ISO8583-65	Activity limit exceeded.	Plafond dépassé	[illegible]	Bank rejection.
82	ISO8583-82	CVV Failure or CVV Value supplied is not valid.	Echec CVV ou valeur CVV fournie non valide.	[illegible]	Bank rejection.
86	ISO8583-86	Payment authorization rejected	Autorisation de paiement rejetée	[illegible]	Bank rejection.
90	ISO8583-90	Payment authorization rejected	Autorisation de paiement rejetée	[illegible]	Bank rejection.
91	ISO8583-91	Payment authorization failed	Autorisation de paiement échouée	[illegible]	Bank interface timeout.
96	ISO8583-96	Payment authorization failed	Autorisation de paiement échouée	[illegible]	Failed operation.

Notice:

- *Web developers often encounter the 3D-1004 error during their first integration tests. In this case, the origin of the error usually comes from the fact that the developer forgets to enter his secret hash key at Merchant Center CMI. The procedure for setting the secret hash key is specified in the CMI integration kit.*