PAYMENT OPERATIONS

SALE request

Payment Platform supports two main operation type: Single Message System (SMS) and Dual Message System (DMS).

SMS is represented by SALE transaction. It is used for authorization and capture at a time. This operation is commonly used for immediate payments.

DMS is represented by AUTH and CAPTURE transactions. AUTH is used for authorization only, without capture. This operation used to hold the funds on card account (for example to check card validity).

SALE request is used to make both SALE and AUTH transactions.

If you want to make AUTH transaction, you need to use parameter auth with value Y.

If you want to send a payment for the specific sub-account (channel), you need to use channel_id, that specified in your Payment Platform account settings.

This request is sent by POST in the background (e.g. through PHP CURL).

Request parameters

Parameter	Description	Values	Required field
action	Sale	SALE	+
client_key	Unique key (client_key)	UUID format value	+
channel_id	Payment channel (Sub-account)	String up to 16 characters	-
order_id	Transaction ID in the Merchants system	String up to 255 characters	+
order_amount	The amount of the transaction	Numbers in the format XXXX.XX Pay attention that amount format depends on currency exponent. If exponent = 0, then amount is integer (without decimals). It used for currencies: CLP, VND, ISK, UGX, KRW, JPY.	+

		If exponent = 3, then format: XXXX.XXX (with 3 decimals). It used for currencies: BHD, JOD, KWD, OMR, TND.	
order_currency	Currency	3-letter code	+
order_description	Description of the transaction (product name)	String up to 1024 characters	+
req_token	Special attribute pointing for further tokenization	Y or N (default N)	-
card_token	Credit card token value	String 64 characters	-
card_number	Credit Card Number		+ *
card_exp_month	Month of expiry of the credit card	Month in the form XX	+ *
card_exp_year	Year of expiry of the credit card	Year in the form XXXX	+ *
card_cvv2	CVV/CVC2 credit card verification code	3-4 symbols	+
payer_first_name	Customer's name	String up to 32 characters	+
payer_last_name	Customer's surname	String up to 32 characters	+
payer_middle_name	Customer's middle name	String up to 32 characters	-
payer_birth_date	Customer's birthday	format yyyy-MM-dd, e.g. 1970-02-17	-
payer_address	Customer's address	String up to 255 characters	+
payer_address2	The adjoining road or locality (if required) of the customer's address	String up to 255 characters	-
payer_country	Customer's country	2-letter code	+
payer_state	Customer's state	String up to 32 characters	-
payer_city	Customer's city	String up to 32 characters	+
payer_zip	ZIP-code of the Customer	String up to 10 characters	+
payer_email	Customer's email	String up to 256 characters	+
payer_phone	Customer's phone	String up to 32 characters	+
payer_ip	IP-address of the Customer	XXX.XXX.XXX.XXX	+
term_url_3ds	URL to which Customer should be returned after 3D-Secure	String up to 1024 characters	+

term_url_target	Name of, or keyword for a browsing context where Customer should be returned according to HTML specification.	String up to 1024 characters Possible values: _blank, _self, _parent, _top or custom iframe name (default _top). Find the result of applying the values in the HTML standard description (Browsing context names)	-
recurring_init	Initialization of the transaction with possible following recurring	Y or N (default N)	-
schedule_id	Schedule ID for recurring payments	String	-
auth	Indicates that transaction must be only authenticated, but not captured	Y or N (default N)	-
parameters	Object that contains extra-parameters required by the acquirer	Format: "parameters": {"param1" : "value1", "param2" : "value2", "param3" : "value3"} See Appendix C for more details	-
hash	Special signature to validate your request to Payment Platform	See Appendix A, Formula 1	+

*****_This field becomes optional if card_token is specified_

If the optional parameter card_token and card data are specified, card_token will be ignored.

If the optional parameters req_token and card_token are specified, req_token will be ignored.

Response parameters

You will get JSON encoded string (see an example on Appendix B) with transaction result. If your account supports 3D-Secure, transaction result will be sent to your Notification URL.

Synchronous mode

Successful sale response

Parameter	Description
action	SALE
result	SUCCESS
status	PENDING / PREPARE / SETTLED; only PENDING when auth = Y
order_id	Transaction ID in the Merchant's system
trans_id	Transaction ID in the Payment Platform
trans_date	Transaction date in the Payment Platform
descriptor	Descriptor from the bank, the same as cardholder will see in the bank statement
recurring_token	Recurring token (get if account support recurring sales and was initialization transaction for following recurring)
schedule_id	Schedule ID for recurring payments. It is available if schedule is used for recurring sale
card_token	If the parameter req_token was enabled Payment Platform returns the token value
amount	Order amount
currency	Currency

Unsuccessful sale response

Parameter	Description
action	SALE
result	DECLINED
status	DECLINED
order_id	Transaction ID in the Merchant's system
trans_id	Transaction ID in the Payment Platform
trans_date	Transaction date in the Payment Platform
descriptor	Descriptor from the bank, the same as cardholder will see in the bank statement
amount	Order amount
currency	Currency
decline_reason	The reason why the transaction was declined

3D-Secure transaction response

Parameter	Description
action	SALE

result	REDIRECT
status	3DS / REDIRECT
order_id	Transaction ID in the Merchant's system
trans_id	Transaction ID in the Payment Platform
trans_date	Transaction date in the Payment Platform
descriptor	Descriptor from the bank, the same as cardholder will see in the bank statement
amount	Order amount
currency	Currency
redirect_url	URL to which the Merchant should redirect the Customer
redirect_params	Object of specific 3DS parameters. It is array if redirect_params have no data. The availability of the redirect_params depends on the data transmitted by the acquirer. redirect_params may be missing. It usually happens when redirect_method = GET
redirect_method	The method of transferring parameters (POST / GET)

Callback parameters

Successful sale response

Parameter	Description
action	SALE
result	SUCCESS
status	PENDING/PREPARE/SETTLED
order_id	Transaction ID in the Merchant's system
trans_id	Transaction ID in the Payment Platform
hash	Special signature, used to validate callback, see Appendix A, Formula 2
recurring_token	Recurring token (get if account support recurring sales and was initialization transaction for following recurring)
schedule_id	It is available if schedule is used for recurring sale
card_token	If the parameter req_token was enabled Payment Platform returns the token value
card	Card mask
card_expiration_date	Card expiration date
trans_date	Transaction date in the Payment Platform
descriptor	Descriptor from the bank, the same as cardholder will see in the bank statement
amount	Order amount
currency	Currency

Unsuccessful sale response

Parameter	Description
action	SALE
result	DECLINED
status	DECLINED
order_id	Transaction ID in the Merchant's system
trans_id	Transaction ID in the Payment Platform
trans_date	Transaction date in the Payment Platform
decline_reason	Description of the cancellation of the transaction
hash	Special signature, used to validate callback, see Appendix A, Formula 2

3D-Secure transaction response

Parameter	Description
action	SALE
result	REDIRECT
status	3DS/REDIRECT
order_id	Transaction ID in the Merchant's system
trans_id	Transaction ID in the Payment Platform
hash	Special signature, used to validate callback, see Appendix A, Formula 2
trans_date	Transaction date in the Payment Platform
descriptor	Descriptor from the bank, the same as cardholder will see in the bank statement
amount	Order amount
currency	Currency
redirect_url	URL to which the Merchant should redirect the Customer
redirect_params	Object with the parameters. It is array if redirect_params have no data. The availability of the redirect_params depends on the data transmitted by the acquirer. redirect_params may be missing. It usually happens when redirect_method = GET
redirect_method	The method of transferring parameters (POST/GET)

CAPTURE request

CAPTURE request is used to submit previously authorized transaction (created by SALE request with parameter auth = Y). Hold funds will be transferred to Merchants account.

This request is sent by POST in the background (e.g. through PHP CURL).

Request parameters

Parameter	Description	Values	Required field
action	Capture previously authenticated transaction	CAPTURE	+
client_key	Unique key (client_key)	UUID format value	+
trans_id	Transaction ID in the Payment Platform	UUID format value	+
amount	The amount for capture. Only one partial capture is allowed	Numbers in the format XXXX.XX Pay attention that amount format depends on currency exponent. If exponent = 0, then amount is integer (without decimals). It used for currencies: CLP, VND, ISK, UGX, KRW, JPY. If exponent = 3, then format: XXXX.XXX (with 3 decimals). It used for currencies: BHD, JOD, KWD, OMR, TND.	-
hash	Special signature to validate your request to payment platform	see Appendix A, Formula 2	+

Response parameters

Synchronous mode

Successful capture response

Parameter	Description
action	CAPTURE
result	SUCCESS
status	SETTLED
amount	Amount of capture
order_id	Transaction ID in the Merchants system
trans_id	Transaction ID in the Payment Platform
trans_date	Transaction date in the Payment Platform
descriptor	Descriptor from the bank, the same as cardholder will see in the bank statement
currency	Currency

Unsuccessful capture response

Parameter	Description
action	CAPTURE
result	DECLINED
status	PENDING
order_id	Transaction ID in the Merchant's system
trans_id	Transaction ID in the Payment Platform
trans_date	Transaction date in the Payment Platform
descriptor	Descriptor from the bank, the same as cardholder will see in the bank statement
amount	Amount of capture
currency	Currency
decline_reason	The reason why the capture was declined

Callback parameters

Successful capture response

Parameter	Description
action	CAPTURE
result	SUCCESS
status	SETTLED
order_id	Transaction ID in the Merchant's system

trans_id	Transaction ID in the Payment Platform
amount	Amount of capture
trans_date	Transaction date in the Payment Platform
descriptor	Descriptor from the bank, the same as cardholder will see in the bank statement
currency	Currency
hash	Special signature, used to validate callback, see Appendix A, Formula 2

Unsuccessful capture response

Parameter	Description
action	CAPTURE
result	DECLINED
status	PENDING
order_id	Transaction ID in the Merchant's system
trans_id	Transaction ID in the Payment Platform
decline_reason	The reason why the capture was declined
hash	Special signature, used to validate callback, see Appendix A, Formula 2

CREDITVOID request

CREDITVOID request is used to complete both REFUND and REVERSAL transactions.

REVERSAL transaction is used to cancel hold from funds on card account, previously authorized by AUTH transaction.

REFUND transaction is used to return funds to card account, previously submitted by SALE or CAPTURE transactions.

This request is sent by POST in the background (e.g. through PHP CURL).

Request parameters

Parameter	Description	Values	Required field
action	CREDITVOID	CREDITVOID	+
client_key	Unique key (client_key)	UUID format value	+
trans_id	Transaction ID in the Payment Platform	UUID format value	+
amount	The amount of full or partial refund. If amount is not specified, full refund will be issued. In case of partial refund this parameter is required. Several partial refunds are allowed	Numbers in the format XXXX.XX Pay attention that amount format depends on currency exponent. If exponent = 0, then amount is integer (without decimals). It used for currencies: CLP, VND, ISK, UGX, KRW, JPY. If exponent = 3, then format: XXXX.XXX (with 3 decimals). It used for currencies: BHD, JOD, KWD, OMR, TND.	-
hash	Special signature to validate your request to Payment Platform	see Appendix A, Formula 2	+

Response parameters

Synchronous mode

Parameter	Description
action	CREDITVOID
result	ACCEPTED
order_id	Transaction ID in the Merchant's system
trans_id	Transaction ID in the Payment Platform

Callback parameters

Successful refund/reversal response#

Parameter	Description
action	CREDITVOID
result	SUCCESS
status	REFUND/REVERSAL - for full refund SETTLED - for partial refund
order_id	Transaction ID in the Merchant's system
trans_id	Transaction ID in the Payment Platform
creditvoid_date	Date of the refund/reversal
amount	Amount of refund
hash	Special signature, used to validate callback, see Appendix A, Formula 2

Unsuccessful refund/reversal response

Parameter	Description
action	CREDITVOID
result	DECLINED
order_id	Transaction ID in the Merchant's system
trans_id	Transaction ID in the Payment Platform
decline_reason	Description of the cancellation of the transaction
hash	Special signature, used to validate callback, see Appendix A, Formula 2

GET_TRANS_STATUS request

Gets order status from Payment Platform. This request is sent by POST in the background (e.g. through PHP CURL).

Request parameters

Parameter	Description	Values	Required field
action	GET_TRANS_STATUS	GET_TRANS_STATUS	+
client_key	Unique key (client_key)	UUID format value	+
trans_id	Transaction ID in the Payment Platform	UUID format value	+
hash	Special signature to validate your request to Payment Platform	CREDIT2CARD - see Appendix A, Formula 6 Others - see Appendix A, Formula 2	+

Response parameters

Parameter	Description
action	GET_TRANS_STATUS
result	SUCCESS
status	3DS / REDIRECT / PENDING / PREPARE / DECLINED / SETTLED / REVERSAL / REFUND / CHARGEBACK
order_id	Transaction ID in the Merchant` s system
trans_id	Transaction ID in the Payment Platform
decline_reason	Reason of transaction decline. It shows for the transactions with the DECLINED status
recurring_token	Token for recurring. It shows when the next conditions are met for the SALE transaction: - transaction is successful - SALE request contained recurring_init parameter with the value 'Y'

GET_TRANS_DETAILS request

Gets all history of transactions by the order. This request is sent by POST in the background (e.g. through PHP CURL).

Request parameters

Parameter	Description	Values	Required field
action	GET_TRANS_DETAILS	GET_TRANS_DETAILS	+
client_key	Unique key (client_key)	UUID format value	+
trans_id	Transaction ID in the Payment Platform	UUID format value	+
hash	Special signature to validate your request to Payment Platform	CREDIT2CARD - see Appendix A, Formula 6 Others - see Appendix A, Formula 2	+

Response parameters#

Parameter	Description
action	GET_TRANS_DETAILS
result	SUCCESS
status	3DS / REDIRECT / PENDING / PREPARE / DECLINED / SETTLED / REVERSAL / REFUND / CHARGEBACK
order_id	Transaction ID in the Merchant` s system
trans_id	Transaction ID in the Payment Platform
name	Payer name
mail	Payer mail
ip	Payer IP
amount	Order amount
currency	Currency
card	Card in the format XXXXXX****XXXX
decline_reason	Reason of transaction decline.It shows for the transactions with the DECLINED status
recurring_token	Token for recurring. It shows when the next conditions are met for the SALE transaction: - transaction is successful - SALE request contained recurring_init parameter with the value 'Y' - SALE request contained card data which was used for the first time
schedule_id	Schedule ID for recurring payments
transactions	Array of transactions with the parameters: - date - type (sale, 3ds, auth, capture, credit, chargeback, reversal, refund) - status (success, waiting, fail) - amount

GET_TRANS_STATUS_BY_ORDER request

Gets the status of the most recent transaction in the order's transaction subsequence from Payment Platform. This request is sent by POST in the background (e.g. through PHP CURL).

It is recommended to pass an unique order_id in the payment request. That way, it will be easier to uniquely identify the payment by order_id. This is especially important if cascading is configured. In this case, several intermediate transactions could be created within one payment.

Request parameters#

Parameter	Description	Values	Required field
action	GET_TRANS_STATUS_BY_ORDER	GET_TRANS_STATUS_BY_ORDER	+
client_key	Unique key (client_key)	UUID format value	+
order_id	Transaction ID in the Merchants system	UUID format value	+
hash	Special signature to validate your request to Payment Platform	see Appendix A, Formula 7	+

Response parameters#

Parameter	Description
action	GET_TRANS_STATUS_BY_ORDER
result	SUCCESS
status	3DS / REDIRECT / PENDING / PREPARE / DECLINED / SETTLED / REVERSAL / REFUND / CHARGEBACK
order_id	Transaction ID in the Merchant` s system
trans_id	Transaction ID in the Payment Platform
decline_reason	Reason of transaction decline. It shows for the transactions with the DECLINED status
recurring_token	Token for recurring. It shows when the next conditions are met for the SALE transaction: - transaction is successful - SALE request contained recurring_init parameter with the value 'Y'

RECURRING_SALE request

Recurring payments are commonly used to create new transactions based on already stored cardholder information from previous operations.

RECURRING_SALE request has same logic as SALE request, the only difference is that you need to provide primary transaction id, and this request will create a secondary transaction with previously used cardholder data from primary transaction.

This request is sent by POST in the background (e.g. through PHP CURL).

Request parameters#

Parameter	Description	Values	Required field
action	Recurring sale	RECURRING_SALE	+
client_key	Unique key (CLIENT_KEY)	UUID format value	+
order_id	Transaction ID in the Merchant's system	String up to 255 characters	+
order_amount	The amount of the transaction	Numbers in the format XXXX.XX Pay attention that amount format depends on currency exponent. If exponent = 0, then amount is integer (without decimals). It used for currencies: CLP, VND, ISK, UGX, KRW, JPY. If exponent = 3, then format: XXXX.XXX (with 3 decimals). It used for currencies: BHD, JOD, KWD, OMR, TND.	+
order_description	Transaction description (product name)	String up to 1024 characters	+
recurring_first_trans_id	Transaction ID of the primary transaction in the Payment Platform	UUID format value	+

recurring_token	Value obtained during the primary transaction	UUID format value	+
schedule_id	Schedule ID for recurring payments	String	-
auth	Indicates that transaction must be only authenticated, but not captured	Y or N (default N)	-
hash	Special signature to validate your request to payment platform	see Appendix A, Formula 1	+

Response parameters

Response from Payment Platform is the same as by SALE command, except for the value of the difference parameter

action = RECURRING_SALE. You will receive a JSON encoded string with the result of the transaction.

CHARGEBACK notification parameters

CHARGEBACK transactions are used to dispute already settled payment.

When processing these transactions Payment Platform sends notification to Merchant` s Notification URL.

Parameter	Description
action	CHARGEBACK
result	SUCCESS
status	CHARGEBACK
order_id	Transaction ID in the Merchant` s system
trans_id	Transaction ID in the Payment Platform
amount	The amount of the chargeback
chargeback_date	System date of the chargeback
bank_date	Bank date of the chargeback
reason_code	Reason code of the chargeback
hash	Special signature to validate callback, see Appendix A, Formula 2

Appendix A (Hash)

Hash - is signature rule used either to validate your requests to payment platform or to validate callback from payment platform to your system. It must be md5 encoded string calculated by rules below:

Formula 1:

hash is calculated by the formula:

md5(strtoupper(strrev(email).PASSWORD.strrev(substr(card_number,0,6).substr(card_number,-4))))

if parameter card_token is specified hash is calculated by the formula:

md5(strtoupper(strrev(email).PASSWORD.strrev(card_token)))

Formula 2:

hash is calculated by the formula:

md5(strtoupper(strrev(email).PASSWORD.trans_id.strrev(substr(card_number,0,6).substr(card_number,-4))))

Formula 3:

hash for **Create a schedule** is calculated by the formula:

md5(strtoupper(strrev(PASSWORD)));

Formula 4:

hash for **Other schedules** is calculated by the formula:

md5(strtoupper(strrev(schedule_id + PASSWORD)));

Formula 5:

hash for **CREDIT2CARD request** is calculated by the formula:

md5(strtoupper(PASSWORD.strrev(substr(card_number,0,6).substr(card_number,-4))))

if card_token is specified hash is calculated by the formula:

md5(strtoupper(PASSWORD. strrev(card_token)))

Formula 6:

hash is calculated by the formula:

md5(strtoupper(PASSWORD.trans_id.strrev(substr(card_number,0,6).substr(card_number,-4))))

Formula 7:

hash is calculated by the formula:

md5(strtoupper(strrev(email).PASSWORD.order_id.strrev(substr(card_number,0,6).substr(card_number,-4))))