

Integration of Google Pay™ Direct API via Unicorn Laboratories (Web)

1. General Information

This document describes the integration of Google Pay Web with the direct H2H API of Unicorn Laboratories for merchants integrated with Unicorn Laboratories as a payment gateway (PSP).

In this document, the term Direct API refers to a merchant's direct H2H integration with the Unicorn Laboratories API. The Google Pay Web API uses the PAYMENT_GATEWAY tokenization type rather than DIRECT: the encrypted token is passed by the merchant to Unicorn Laboratories without decryption. When registering a site in the Google Pay & Wallet Console, the Gateway integration type must be selected.

This document describes:

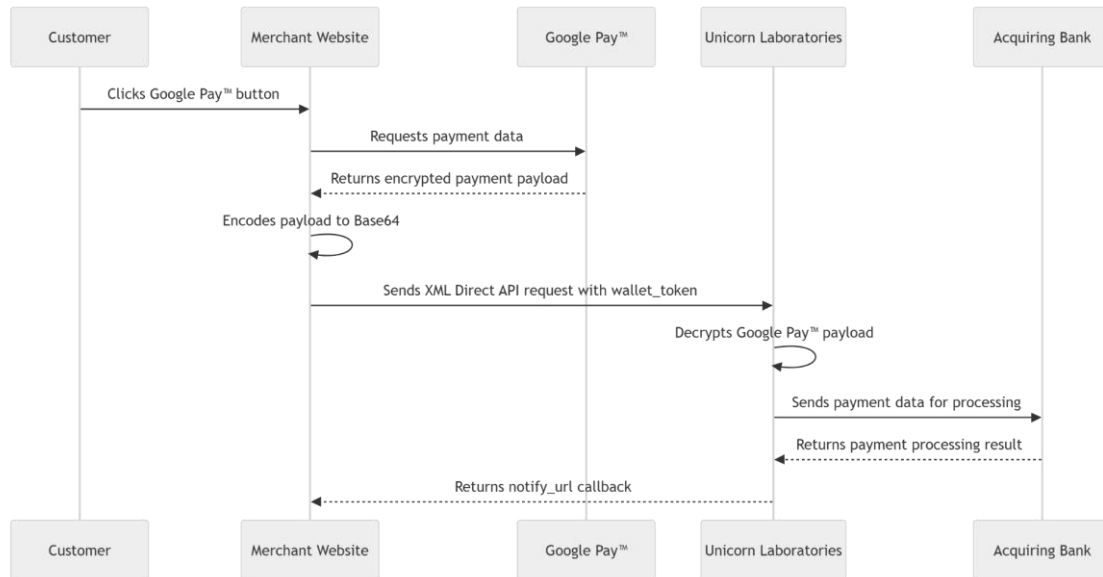
- integration of the Google Pay button on the merchant side;
- integration of the Google Pay Web API;
- Direct API integration with Unicorn Laboratories;
- transmission of the encrypted Google Pay payload to Unicorn Laboratories for decryption and processing.

This document does not describe integration with Unicorn Laboratories Hosted Checkout.

2. Integration Flow

The integration works as follows:

- The customer clicks the Google Pay button on the merchant's website.
- Google Pay returns an encrypted payment payload.
- The merchant receives the encrypted payload from Google Pay.
- The merchant encodes the payload in Base64.
- The merchant sends the payload to Unicorn Laboratories via a Direct API XML request.
- Unicorn Laboratories forwards the payment data to the acquiring bank.
- The acquiring bank returns the payment processing result to Unicorn Laboratories.
- Unicorn Laboratories returns a synchronous response confirming receipt of the operation. The merchant obtains the further result via a status request and notifications to notify_url.



3. Merchant Requirements

The merchant must:

- comply with PCI DSS requirements and other security requirements set out in the agreement and the connection procedure for the Unicorn Laboratories H2H API;
- use HTTPS/TLS;
- use only official Google Pay assets;
- comply with Google Pay branding requirements;
- comply with the Google Pay API Terms of Service;
- comply with the Google Pay API Acceptable Use Policy.

Google documentation:

- [Google Pay Web Documentation](#)
- [Google Pay Web Integration Checklist](#)
- [Google Pay Web Brand Guidelines](#)
- [Google Pay API Terms of Service](#)
- [Google Pay API Acceptable Use Policy](#)

4. Supported Authorization Methods

Unicorn Laboratories supports the following Google Pay authorization methods:

Method	Description	3DS
PAN_ONLY	Google Pay returns the card PAN	Standard risk-based 3DS logic is required
CRYPTOGRAM_3DS	Network/device token with a cryptogram	Additional 3DS is usually not required

Important

For operations using the PAN_ONLY authorization method, Unicorn Laboratories passes the decrypted card data to the acquiring bank. The bank applies standard risk-based checks and initiates 3-D Secure if necessary. The merchant does not perform 3-D Secure itself. If the operation has not yet completed (final="0"), the merchant must request the payment status. If the auth.url attribute is present, the customer must be redirected to that URL. After passing 3-D Secure, the result is delivered via notify_url. The current final status can also be obtained by requesting status.

Status processing algorithm: final="0" with auth.url present — redirect the customer to auth.url; final="0" without auth.url — retry the status request later; final="1" — process the final result.

Obtaining the 3DS link via a payment status request:

```
<?xml version="1.0" encoding="UTF-8"?>
<request point="4354">
  <status id="70000167"/>
</request>
```

The response returns the auth.url parameter, to which the customer must be redirected to complete 3DS:

```
<response>
  <result id="70000167" state="40" substate="3" code="0" final="0" trans="70000167"
sum_prov="10500" server_time="2026-05-12T12:32:51+0500">
    <attribute name="country_code" value="KZ"/>
    <attribute name="redirect_success_url" value="https://merchant.example/success"/>
    <attribute name="amount_currency" value="398"/>
    <attribute name="card_pan" value="489993*****4361"/>
    <attribute name="redirect_fail_url" value="https://merchant.example/fail"/>
    <attribute name="notify_url" value="https://merchant.example/callback"/>
    <attribute name="email" value="customer@example.com"/>
    <attribute name="nonce" value="40E667E1DFC6DD09494A2D008B3F4941"/>
    <attribute name="auth.url" value="https://auth.unicornlab.kz/pay/70000167"/>
  </result>
</response>
```

To complete 3-D Secure, the merchant uses only auth.url. Internal fields used in the interaction between Unicorn Laboratories and the ACS are not part of the public contract.

5. Supported Payment Systems

Supported payment systems:

- VISA
- MASTERCARD

Example Google Pay configuration:

```
allowedCardNetworks: ["MASTERCARD", "VISA"]
```

Documentation:

6. Google Merchant ID

Merchants must obtain their Google Merchant ID through the Google Pay & Wallet Console: [Google Pay & Wallet Console](#)

When adding a site to the Google Pay & Wallet Console, the merchant must select the Gateway integration type.

A Google Merchant ID is required for production Google Pay Web integrations, and its configuration must be:

```
merchantInfo.merchantId
```

Example:

```
merchantInfo: {  
  merchantId: "123456789012345",  
  merchantName: "ABC Company"  
}
```

Important

Google Merchant ID:

- is issued by Google;
- is used for production Google Pay web integration;
- is specified in merchantInfo.merchantId;
- is not the same as gatewayMerchantId.

Note: the Google Merchant ID is used by the merchant only in the client-side configuration of the Google Pay Web API (merchantInfo.merchantId). It is passed to Google Pay and is not sent in the Unicorn Laboratories XML request.

7. Gateway Configuration

Merchants integrated via Unicorn Laboratories must use the following Google Pay tokenization configuration:

```
tokenizationSpecification: { type: "PAYMENT_GATEWAY",  
  parameters: {  
    gateway: "unicornlab", gatewayMerchantId: "unicorn"  
  }  
}
```

Important: for this integration, gatewayMerchantId must have the value "unicorn".

8. Obtaining the Google Pay Token

After successful authorization, Google Pay returns an encrypted payload:

```
paymentData.paymentMethodData.tokenizationData.token
```

The merchant must:

- obtain the encrypted token;
- encode the UTF-8 bytes of the resulting token string using standard Base64 per RFC 4648, with padding and no line breaks;
- pass the result in `wallet_token` without parsing or reassembling the original JSON.

9. Payment API Request

Request format:

- XML
- UTF-8

Host: <https://payment.unicornlab.kz/external/extended-cert>

Example request

```
<?xml version="1.0" encoding="UTF-8"?>
<request point="4354">
  <payment id="{{counter_id}}" sum="3000" check="0" service="1330" date="{{current_date}}"
  account="">
    <attribute name="amount_currency" value="398"/>
    <attribute name="notify_url" value="https://merchant.example/callback"/>
    <attribute name="redirect_success_url" value="https://merchant.kz/success"/>
    <attribute name="redirect_fail_url" value="https://merchant.kz/fail"/>
    <attribute name="wallet_token" value="BASE64_TOKEN"/>
    <attribute name="pay_method" value="google_pay"/>
    <attribute name="email" value="customer@example.com"/>
    <attribute name="country_code" value="KZ"/>
    <attribute name="ip" value="203.0.113.10"/>
    <attribute name="desc" value="Payment description"/>
  </payment>
</request>
```

Note: `{{current_date}}` is a Postman variable. Before sending, it must be replaced with the actual date and time in the format `yyyy-MM-dd'T'HH:mm:ssZ`, for example `2026-09-02T12:00:00+0500`.

Description of Payment Request Fields

Field	Required	Format	Length	Example	Description
id	Yes	string	6-32	12345678	Unique merchant payment identifier

Field	Required	Format	Length	Example	Description
sum	Yes	integer	25	3000	Amount in minimum units (1 tenge = 100 tiyn)
check	No	integer	1–5	0	Receipt number. If absent, non-numeric, or greater than 32767, 0 is used. This field is not passed in the request to BCC; for consistency, it is recommended to specify check="0".
service	Yes	numeric	10	1330	Service identifier
date	Yes	yyyy-MM-dd'T'HH:mm:ssZ	24	2026-09-02T12:00:00+0500	Mandatory payment date and time with a UTC offset. If the attribute is missing or has an invalid format, the request will not be parsed.
account	Yes	string	16	""	The attribute must be present; for Google Pay, an empty value should be passed. After decryption, Unicorn Laboratories replaces the input value with a masked PAN/DPAN with a Google Pay suffix for displaying the operation.
amount_currency	Yes	ISO 4217	3	398	Currency code
notify_url	No	HTTPS URL	1024	https://merchant.kz/callback	Notification URL; must be passed to receive the asynchronous result
redirect_success_url	Conditional	HTTPS URL	1024	https://merchant.kz/success	Customer return URL after successful 3-D Secure
redirect_fail_url	Conditional	HTTPS URL	1024	https://merchant.kz/fail	Customer return URL after unsuccessful 3-D Secure
wallet_token	Yes	Base64	65535	BASE64_TOKEN	Encrypted Google Pay payload
pay_method	Yes	google_pay	10	google_pay	Payment method
email	Yes	email	254	customer@example.com	Customer email; passed to BCC in the EMAIL field. If absent, BCC receives an empty value.
country_code	No	ISO 3166-1 alpha-2	2	KZ	Customer country code. In the current Google Pay implementation, this is not passed in the request to BCC; the field may be omitted.
ip	Yes	IPv4 / IPv6	45	203.0.113.10	The customer's actual IP address; passed to BCC in the CLIENT_IP field and used in the risk/3-D Secure process. The IP address of the backend server or reverse proxy should not be passed.
desc	Yes	UTF-8	125	Payment	Payment description; must be passed by the merchant and is forwarded to BCC in the DESC field.

10. Callback Notifications

Once the operation reaches a final status, Unicorn Laboratories sends a notification to `notify_url` via two HTTP requests: GET and POST. This behavior is retained for backward compatibility with previously connected merchants. Both requests relate to the same status-change event; most parameters are the same, while POST additionally contains `account`, `point_id`, and `sum`.

11. Payment Status Request

The payment status can be obtained via a status request. Request format:

- XML
- UTF-8

Host: `https://payment.unicornlab.kz/external/extended-cert`

Example payment status request

```
<?xml version="1.0" encoding="UTF-8"?>
<request point="4354">
  <status id="70000175"/>
</request>
```

Example response to a payment status request

The set of additional attributes depends on the acquiring bank's response and the Unicorn Laboratories configuration. Bank-internal service attributes are not part of the mandatory API contract. For business processing, the result attributes `id`, `state`, `substate`, `code`, `final`, and `trans` should be used. The `pg_error_description` attribute may be passed with the error text.

```

<response>
  <result id="70000175" state="60" substate="0" code="0" final="1" trans="70000175"
sum_prov="10500" server_time="2026-05-12T15:37:37+0500" process_time="2026-05-
12T15:37:42+0500">
    <attribute name="country_code" value="KZ"/>
    <attribute name="card_pan" value="520424*****1234"/>
    <attribute name="pay_method" value="google_pay"/>
    <attribute name="redirect_success_url" value="https://merchant.example/success"/>
    <attribute name="amount_currency" value="398"/>
    <attribute name="redirect_fail_url" value="https://merchant.example/fail"/>
    <attribute name="notify_url" value="https://merchant.example/callback"/>
    <attribute name="email" value="customer@example.com"/>
    <attribute name="TERMINAL" value="EXAMPLE_TERMINAL"/>
    <attribute name="CVC2_RC" value="0"/>
    <attribute name="CARD_MASK" value="520424XXXXXX1234"/>
    <attribute name="APPROVAL" value="EXAMPLE_APPROVAL"/>
    <attribute name="LANG" value="ru"/>
    <attribute name="RC_TEXT" value="Completed successfully"/>
    <attribute name="INT_REF" value="EXAMPLE_INT_REF"/>
    <attribute name="DESC" value="payment"/>
    <attribute name="RRN" value="EXAMPLE_RRN"/>
    <attribute name="MERCHANT" value="EXAMPLE_MERCHANT"/>
    <attribute name="ORDER" value="70000175"/>
    <attribute name="MERCH_TOKEN_ID" value="EXAMPLE_TOKEN_REFERENCE"/>
    <attribute name="RC_TEXT_DECODE" value="Completed successfully"/>
    <attribute name="AMOUNT" value="105.00"/>
    <attribute name="TIMESTAMP" value="20260512104038"/>
    <attribute name="CURRENCY" value="398"/>
    <attribute name="TRAN_AMOUNT" value="105.00"/>
    <attribute name="P_SIGN" value="EXAMPLE_SIGNATURE"/>
    <attribute name="MERCH_RN_ID" value="EXAMPLE_MERCH_RN_ID"/>
    <attribute name="NAME" value="TEST CUSTOMER"/>
    <attribute name="RC" value="00"/>
    <attribute name="MERCH_GMT" value="+5"/>
    <attribute name="ACTION" value="0"/>
    <attribute name="BACKREF" value="https://merchant.example/result"/>
    <attribute name="id1" value="70000175"/>
    <attribute name="TRAN_CUR_NAME" value="KZT"/>
    <attribute name="TRTYPE" value="1"/>
    <attribute name="NONCE" value="EXAMPLE_NONCE"/>
  </result>
</response>

```

In the final response with final="1", 3-D Secure service fields are not used in the public example. For 3-D Secure, the merchant works only with auth.url from the intermediate response with final="0".

12. Requirements for wallet_token

wallet_token must:

- contain the full token string from paymentData.paymentMethodData.tokenizationData.token;

- be encoded as standard Base64 (RFC 4648) from the UTF-8 bytes of this string, with padding and no line breaks;
- be passed without parsing or reassembling the JSON.

The merchant is prohibited from:

- changing the payload structure;
- removing payload fields;
- logging the decrypted PAN;
- logging the cryptogram in plaintext;
- storing sensitive authentication data.

13. Billing Address

Billing address is optional and may be enabled by the merchant if needed for internal anti-fraud logic or payment processing.

```
billingAddressRequired: true
```

Example:

```
billingAddressParameters: {
  format: "MIN",
  phoneNumberRequired: true
}
```

Documentation:

[BillingAddressParameters](#)

14. Requirements for Google Pay UI and Branding

The merchant must:

- use only official Google Pay buttons;
- not modify Google Pay assets;
- comply with Google Pay spacing requirements;
- comply with Google Pay UX guidelines.

Documentation:

- [Google Pay Web Brand Guidelines](#)
- [Google Pay Button Assets](#)

15. Testing Requirements

Before going live in production, the merchant must:

- test PAN_ONLY transactions;
- test CRYPTOGRAM_3DS transactions;
- test success/failure scenarios;

- test webhook processing;
- verify the redirect URLs;
- verify the display of the Google Pay button.

Documentation:

[Google Pay Integration Checklist](#)

16. Production Access

To request production access for Google Pay Web, merchants must follow Google's official guide:

In the Google Pay & Wallet Console, the Gateway integration type must be selected for the site.

[Publish Your Google Pay Web Integration](#)

17. Payment Cancellation and Refund

The cancel operation is used for a full or partial cancellation/refund. For a two-stage payment, prior to capture, the hold amount is cancelled. After capture, as well as for a single-stage payment, a refund of the previously captured amount is performed.

Cancellation of a preliminary authorization for cards processed by the BCC acquirer:

for the full amount, it is performed instantly in favor of the cardholder

for a partial amount, it is performed on the next day in favor of the cardholder after clearing has been processed.

In practice, it is convenient for the merchant to first perform a partial clearing, and the cancellation of the remaining held amount for cards processed by the BCC acquirer is carried out by the bank automatically on the next day in favor of the cardholder after clearing has been processed.

A full or partial refund is possible for a single-stage payment and for a two-stage payment after capture (clearing).

When sending the request, the payment@id value of the original payment and the cancel operation must be used.



Full Refund / Payment Cancellation

REQUEST:

```
<?xml version="1.0" encoding="UTF-8"?>
<request point="4354">
  <cancel id="54827360" sum="2000"/>
</request>
```

RESPONSE:

```
<response>
  <result id="54827360" state="0" sum="2000" request="17223"></result>
</response>
```

Partial Refund / Payment Cancellation

REQUEST:

```
<?xml version="1.0" encoding="UTF-8"?>
<request point="4354">
  <cancel id="53725368" sum="100"/>
</request>
```

RESPONSE:

```
<response>
  <result id="53725368" state="0" sum="100" request="17224"></result>
</response>
```

To find out the final status of the cancellation/refund, the following request must be sent with the request attribute and its value from the previous response.

REQUEST:

```
<?xml version="1.0" encoding="UTF-8"?>
<request point="4354">
  <cancel id="53725368" sum="100" request="17224"/>
</request>
```

18. Notify_url (status callback)

After transaction processing is complete, a callback with the payment result is sent to the previously specified `notify_url`.

Unicorn Laboratories sends two incoming notifications for the same operation result: GET with parameters in the query string, and POST with an `application/x-www-form-urlencoded` body.

The payment should be identified by `refNo` or `order.id`, and the result by `state` and `status`. For a final payment, both fields have a value of 60 (success) or 80 (error).

Parameter	Method	Example value	Description
refNo	GET, POST	jEbxNiPqW	Identifier of the merchant's original payment
order.id	GET, POST	54848485	Internal Unicorn Laboratories operation identifier
id	GET, POST	jEbxNiPqW	Additional identifier from the operation attributes; its presence depends on the configuration
state	GET, POST	60	Transaction state code
status	GET, POST	60	Payment status code
type	GET, POST	payment	Operation type: payment or refund
pg_error_description	GET, POST	Completed successfully	Text description of the result; may be absent or empty
point_id	POST	4438	Point (outlet) identifier
sum	POST	5000	Amount in the minimum currency units
account	POST	520424*****1234 Google Pay	Masked PAN/DPAN with a Google Pay suffix, generated by Unicorn Laboratories after decrypting the token; the merchant's input value is replaced

The actual set of additional attributes depends on the configuration and the bank's result. The merchant should not build processing logic around undocumented bank fields.

19. Error Codes

Payment error codes:

Error code	Description
-100	Duplicate attributes were passed
-2	Payment instrument error
1	Incorrectly specified number
3	Amount outside the allowed range
7	General provider error
8	Insufficient funds
9	Invalid processing parameters
30	Insufficient funds to add the payment
33	Service unavailable for this agent
1008	Payment blocked by a fraud scenario

Online verification error codes:

Error code	Description
-100	Duplicate attributes passed
1000	Invalid number
1001	Provider's server unavailable
1002	Payment acceptance prohibited by the provider
1003	Verification through the provider's server is unavailable. You may proceed with the payment if you are confident the parameters are correct
1004	No outstanding debts found
1005	Payment redirected to another service
1006	The provider cannot perform the verification, but you may proceed with the payment if you are confident the details are correct
1007	Payment is not possible because the operator does not have the personal data

Error code	Description
1008	Rejected by the fraud monitoring system. This check is performed during online verification if, in the "Reference Data — System Parameters" section, on the "Fraud Monitoring System" tab, the "Use blacklist during online verification" checkbox is enabled
1100	Insufficient funds in the merchant's balance to execute the request

Status and substatus codes:

Status	Substatus	Final	Description
0	0	No	New. The status in which payments are inserted into the database
0	1	No	Ready for processing. Preparation for processing and queuing
0	2	No	Determining the provider according to the payment routing rules
0	3,4	No	Fraud control. Payment verification by the fraud monitoring system
0	5	No	Confirmation. The status in which payments are loaded into the merchant portal via the "Dispatch — Payment Upload" section. Payments remain in this status until the user confirms processing (the "Process all deferred" button)
0	6	No	Provider not set. Payments for services for which the processing route is not defined fall into this status
0	7	No	Timeout. The payment waits for the processing deferral period set in the service settings
0	8	No	Deferred. Payment status when the provider setting "Use deferred processing" is enabled. Payments remain in this status until manually processed
0	9	No	Awaiting confirmation. A payment from an external system with the delayed flag receives this status; the payment remains in it until confirmation is received
0	11	No	Reward not set. Payments for services that have not been added to the reward profile fall into this status
10	Any	No	Payment blocked (as a result of applying fraud monitoring rules)
20	1	No	Ready for capture

Status	Substatus	Final	Description
20	2	No	Batch capture
20	3	No	Single capture
20	4	No	Capture error
20	5	No	Waiting for payment instrument
30	Any	No	Preliminary payment verification
30	1	No	Ready for preliminary verification (by pre-check modules)
30	2	No	Preliminary verification, in progress
30	3	No	Preliminary verification ended with an ambiguous error. After the global timeout expires, the operation is moved to status 40/4 to release the reserve, and then to the final status 80/5
30	4	No	Blocked, manual investigation required
30	5	No	Blocked, rejected
40	1	No	Ready for processing
40	2,3	No	Processing
40	4,5,6,7	No	Processing error
40	8	No	Awaiting response from external provider
40	9	No	Waiting for confirmation from external system
40	10	No	Waiting for missing attributes required to complete processing (for operations in a gap status)
60	0	Yes	Successful payment processing status (final): holding and/or capture of funds
80	Any	Yes	Error status (final)
80	1,2,3	Yes	Payment cancelled manually
80	4	Yes	Insufficient funds in the account
80	5	Yes	Processing error
80	6,7,8	Yes	Other error
80	9	Yes	Refund
-1	0	No	Error inserting the operation into the database (not final)

Status	Substatus	Final	Description
-2	0	Yes	Payment not found when checking status (final)

Payment Operation Statuses

Status code	Description of the payment status in processing	Final status
0	Operation added (status — New)	false
20	Operation is being processed (status — Financial control)	false
40	Operation is being processed (status — Processing)	false
60	Operation successfully processed (status — Success)	true
80	Operation in error (status — Error)	true
-2	Payment not found when checking status	true

Cancellation Request Processing Status Codes

Status	Final	Description
-1	Yes	User not found
-2	Yes	The user belongs to a different merchant (not the same as the outlet's)
-3	Yes	User is blocked
-4	Yes	Payment not found
-5	Yes	Payment is not in success (in an incorrect status)
-6	Yes	Invalid cancellation amount (may exceed the payment amount)
-10	Yes	Unknown error (returned if an exception occurred during processing)
0	No	New request status
1	No	In progress
2	Yes	Completed (Success)
3	Yes	Rejected
10	No	Cancellation via gateway (the cancellation is processed by the provider's gateway, not the processing center manager)
11	No	Cancellation via gateway, processing

Status	Final	Description
12	No	Cancellation via gateway, unknown
13	Yes	Cancellation via gateway, success
14	Yes	Cancellation via gateway, error

Error numbers for the service attribute in an advanced request

Error number	Description
0	Success
1	No data found for the specified details
2	Payment of the specified number is prohibited by the service provider
3	Unable to parse the provider's response. Please try again later
4	Online requests are not implemented for this service
5	No outstanding debts
6	One of the arguments is missing or invalid
7	Too many items match the specified search criteria. Please refine the request parameters
8	No records found matching the specified search criteria. Please change the search criteria
9	A certain error with a text description. The text is returned in the element <code><input key="message" value="Card recovery is required"/></code>
10	Verification could not be performed. Decide independently whether to continue or stop payment processing
11	Payment is not possible because the operator does not have the personal data
100	Insufficient funds in the balance to execute the request