IPAY

API Documentation

Table of contents

List of images

Version

Date	Details
12.05.2021	Adding the email parameter to the register.do/registerPreAuth.do API call
06.08.2021	Partially refund
25.11.2021	Payment in installments and/or in loyalty points
17.01.2022	Adding error codes - actionCode 804 and 913
22.02.2022	Adding error codes - actionCode 104, 124 and 952
11.03.2022	Adding response parameters for the getOrderStatusExtended.do API - refunds and chargeback
03.10.2022	orderBundle requirements
22.11.2022	Data types used when calling APIs
23.11.2022	Adding the sessionTimeoutSecs and expirationDate parameters
08.02.2023	Additional information for actionCode 803, 804 and 913
21.02.2023	Adding getFinishedPaymentInfo.do API
26.05.2023	Additional specifications for the description parameter in the register.do/registerPreAuth.do API call
27.07.2023	Adding errorCode 11 to the register.do/registerPreAuth.do API call
22.11.2023	Additional specifications for the description parameter in the register.do/registerPreAuth.do API call
22.03.2024	Modify host on test environment. Additional specifications for the returnUrl parameter in the register.do/registerPreAuth.do API call. Highlight transactions with exception in iPAY console. Additional information for merchantOrderParams and attributes in the getOrderStatusExtended.do API call. API Endpoint Authentication.
06.09.2024	Update checkout page images
25.11.2024	Added response parameters for the getOrderStatusExtended.do API – bindingInfo – for cards saved using network token technology
12.02.2026	Change the transaction limit in loyalty points (loy) from 600 to 1500

20.02.2026	Add parameter to getOrderStatusExtended.do API call - includeCardArt Add response parameters for getOrderStatusExtended.do API - paymentWay Add possible responses for errorMessage parameter in deposit.do / refund.do / reverse.do API Add parameter to deposit.do API call - depositLoyalty Add response parameters for deposit.do API - loyaltyOperationActionCode, loyaltyOperationActionCodeDescription Add parameter to reverse.do API call - reverseLoyalty Add response parameters for reverse.do API - loyaltyOperationActionCode Add parameter to refund.do API call - refundLoyalty Add response parameters for refund.do API - loyaltyOperationActionCode, loyaltyOperationActionCodeDescription

1 Introduction

The iPay system has two types of payments, depending on the services offered by the merchant.

1.1 Payments by 1-Phase

1-Phase is suitable for services such as insurance, tickets / subscriptions, where "deposit" is made automatically for successfully completed transactions.

The collection of transactions made with 1-phase and approved (status "Deposited"), is done automatically, the money is transferred to the account in T + 1 / T + 2 working days - no intervention is required from the merchant.

1.2 Payments by 2-Phase

2-Phase is suitable for goods services that require delivery, where the amount is blocked first (pre-authorization), and upon delivery / confirmation of delivery of goods is made "completion" of the transaction and must be generated "deposit" (collection).

The collection of transactions made with 2-phase, takes place only after the merchant performs the collection operation (by calling the deposit.do API or manually from the iPay console) and for transactions that, following this operation, have the status "Deposited”. The money is transferred to the account in T + 1 / T + 2 working days from the date of the collection operation (deposit).

Attention! It is necessary that the collection operation be carried out within a maximum of 5 days from the date of the transaction, otherwise there is a risk that it can no longer be sent for settlement. This operation can be performed **either for the entire pre-authorized amount or for a lower amount.**

Attention! If the collection operation is performed for an amount lower than the pre-authorized amount, the difference is automatically unlocked on the customer's account.

Attention! If you do not honor the order, please complete the **reverse operation** within a **maximum of 24 hours from the date of the transaction.**

2 Credentials

The credentials are pairs of the username - password type and are provided by Banca Transilvania at the time of creating the merchant on the iPay platform. These are required to access the iPay console and to call APIs.

The credentials are of two types:

- **GUI** (Graphical User Interface) - for accessing the graphical interface - iPay console - which displays the history of transactions made on a merchant (amount, date and time of transaction, type of transaction, transaction status, etc.), as well as the ability to make deposit / reverse / refund directly from the console
- **API** (Application Programming Interface) - for accessing (calling) API services to the payment page provided by Banca Transilvania.

3 Test System Vs. production system

The differences between the test system and the production system are:

- **Production credentials do not work on the test environment or vice versa.**
- Console
    - Test environment - https://ecclients-sandbox.btrl.ro/console/index.html#login
    - Production environment - https://ecclients.btrl.ro/console/index.html#login
- API
    - For payments by 1-Phase
        - Test environment - https://ecclients-sandbox.btrl.ro/payment/rest/register.do + mandatory parameters
        - Production environment - https://ecclients.btrl.ro/payment/rest/register.do + mandatory parameters
    - For payments by 2-Phase
        - Test environment - https://ecclients-sandbox.btrl.ro/payment/rest/registerPreAuth.do + mandatory parameters
        - Production environment - https://ecclients.btrl.ro/payment/rest/registerPreAuth.do + mandatory parameters
- Money is not collected on the test medium
- On the test environment you can only use cards provided by us

In the event of connectivity issues on the test environment, it is required you check that your server can communicate with the host located at ecclients-sandbox.btrl.ro, port 443 .

4 Transaction status and meaning of colors

4.1 For transactions made through 1-Phase

Status	Description	Color
CREATED	The cardholder is redirected to the payment page, but has not completed the payment: either reached the payment page, filled in or did not complete the card details and closed, or reached the payment confirmation page and did not complete the transaction	-
DEPOSITED	The cardholder has entered the card details, confirmed the payment and the transaction is completed successfully	Green - Transaction full 3Dsecure (challenge sau frictionless)
		Yellow - Transaction attempted
		Orange – Transaction with exception
		Blue – Transaction without 3Dsecure / Transaction with “Pay with BT PAY”
PARTIALLY REFUNDED	Partial refund of the money back to the customer's account. After performing the refund operation, the status changes from DEPOSITED to PARTIALLY REFUNDED	Red
REFUNDED	Full refund of the money back to the customer's account. After performing the refund operation, the status changes from DEPOSITED or from PARTIALLY REFUNDED to REFUNDED	Red
DECLINED	The cardholder entered the card details, but the transaction is declined for various reasons (Card blocked, insufficient funds, limit exceeded, wrong CVV, expired card, cardholder's issuing bank declines the transaction, etc)	Red

4.2 For transactions made through 2-Phase

Status	Description	Color
CREATED	The cardholder is redirected to the payment page, but has not completed the payment: either reached the payment page, filled in or did not complete the card details and closed, or reached the payment confirmation page and did not complete the transaction	-
APPROVED	The cardholder has entered the card details and the transaction is successfully pre-authorized - transaction approved - the money is blocked until the deposit is made on the transaction	Green - Transaction full 3Dsecure (challenge sau frictionless) Yellow - Transaction attempted Orange – Transaction with exception Blue – Transaction without 3DSecure
REVERSED	Cancellation of a payment transaction (refund of money back to the customer's account). This operation can be performed only for transactions made with 2-phase and APPROVED status. The reversal operation aims to cancel the pre-authorization, followed by the change of status from APPROVED to REVERSED. If you do not honor the order, please complete the reverse operation within a maximum of 24 hours from the date of the transaction	Red
DEPOSITED	For transactions made with 2-phase, this status appears only when a deposit is made on the initial transaction. It can be done **either for the entire pre-authorized amount, or for a lower amount.** This operation can be performed only once and aims to unlock the money and send the transaction to settlement. The transaction completion / collection operation is followed by the change of status from APPROVED to DEPOSITED	Green - Transaction full 3Dsecure (challenge sau frictionless) Yellow - Transaction attempted Orange – Transaction with exception Blue – Transaction without 3DSecure
PARTIALLY REFUNDED	Partial refund of the money back to the customer's account. After performing the refund operation, the status changes from DEPOSITED to PARTIALLY REFUNDED	Red
REFUNDED	Full refund of the money back to the customer's account. After performing the refund operation, the status changes from DEPOSITED or from PARTIALLY REFUNDED to REFUNDED	Red

Status	Description	Color
DECLINED	The cardholder entered the card details, but the transaction is declined for various reasons (Card blocked, insufficient funds, limit exceeded, wrong CVV, expired card, cardholder's issuing bank declines the transaction, etc.)	Red

5 Significance of ECI

The ECI (Electronic Commerce Indicator) value means the result of 3Dsecure authentication.

5.1 Transaction made with Visa card

Value	Description
05	Both cardholder and card issuing bank are 3D enabled. 3D card authentication is successful
06	Either cardholder or card issuing bank is not 3D enrolled. 3D card authentication is unsuccessful, in sample situations as: - 3D cardholder not enrolled - Card issuing bank is not 3D Secure ready
07	Authentication is unsuccessful or not attempted. The credit card is either a non-3D card or card issuing bank does not handle it as a 3D transaction

5.2 Transaction made with MasterCard card

Value	Description
02	Both cardholder and card issuing bank are 3D enabled. 3D card authentication is successful
01	Either cardholder or card issuing bank is not 3D enrolled. 3D card authentication is unsuccessful, in sample situations as: - 3D cardholder not enrolled - Card issuing bank is not 3D Secure ready
07	Authentication is unsuccessful or not attempted. The credit card is either a non-3D card or card issuing bank does not handle it as a 3D transaction

6 APIs

6.1 General information

The iPay system has two types of payments, depending on the services offered by the merchant:

• **1-Phase** is suitable for services such as insurance, tickets / subscriptions, where "deposit" is made automatically for successfully completed transactions.

• **2-Phase** is suitable for goods services that require delivery, where the amount is blocked first (pre-authorization), and upon delivery / confirmation of delivery of goods is made "completion" of the transaction and must be generated "deposit" (collection).

To register an order, use the *register.do / registerPreAuth.do* method. This request is designed for the registration of online orders and automatic redirection to the payment page provided by Banca Transilvania (page where the card data is entered), followed by the payment confirmation page (3DSecure code verification and validation page).

For payments made with *1-phase*, by calling the API *register.do*, successful transactions are with “Deposited” status, which means that the money is to be collected in the merchant's account without any intervention from him.

For payments made with *2-phase*, by calling the API *registerPreAuth.do*, the transactions made successfully are with “Approved” status, which means that the customer's money for the placed order is blocked. Upon delivery / confirmation of delivery of goods is made "completion" of the transaction, unblocking the amount related to the transaction, being necessary to implement the *deposit.do* method, otherwise you will not cash the transaction. The collection operation is only available for transactions with APPROVED status and can be performed either for the entire pre-authorized amount or for a lower amount. This operation can only be performed once. If the collection operation is performed for an amount lower than the pre-authorized amount, the difference is automatically unlocked on the customer's account.

To cancel the payment of an order made through 2-phase, it is necessary to use the *reverse.do* method. This functionality is available for a limited period, specified by the bank, up to 24 hours from the date of the transaction. The operation can be performed only once.

If the customer returns one or more products, iPay allows the refund of the money for that order back to the customer's account, by implementing the *refund.do* method.

In order to find out details about the status of a registered order, as well as to show the customer the answer regarding the transaction (approved or declined and the reason for declining the transaction), it is necessary to implement the *getOrderStatusExtended.do* method.

The types of data used in the documentation are:

- Decimal(x) numeric length x (maximum x digits)
- String(x) alphanumeric + special length x (maximum x digits, letters or special characters)
- JSON format json described in each case
- ENUM('val.1'..'val.n') one of the predefined enumerated values

APIs for implementation depending on the type of payments			
API name	**1-phase**	**2phase**	**Description**
register.do	YES	NO	Online order registration and automatic redirection to the payment page, followed by the payment confirmation page
registerPreAuth.do	NO	YES	
deposit.do	NO	YES	Unlocking and collecting pre-authorization orders
refund.do	YES	YES	Partial or full refund of the money back to the customer's account
reverse.do	NO	YES	Cancellation of a payment / release of money
getOrderStatusExtended.do	YES	YES	Necessary to display to the customer the answer regarding the transaction (approved or declined and the reason for declining the transaction), as well as other relevant information to the trader regarding the transaction

6.2 API Endpoint Authentication

For the endpoints that require authentication there are two supported authentication methods. (the description of each endpoint specifies if it requires authentication or not)

6.2.1 Using authorization header - recommended method

The authorization header is built by concatenating the "userName" and "password" using the ":" delimiter. This result must then be encoded in base64.

Example: For username *test_exemplu_API* and password *test_exemplu_parola*, the result is "*test_exemplu_API:test_exemplu_parola*", which encoded in base64 yields the value "*dGVzdF9leGVtcGx1X0FQSTp0ZXN0X2V4ZW1wbHVfcGFyb2xh*".

In order to use this for authentication the following header must be sent:

"Authorization: Basic dGVzdF9leGVtcGx1X0FQSTp0ZXN0X2V4ZW1wbHVfcGFyb2xh"

6.2.2 Using the "userName" and "password" in the body of the request - NOT recommended method

Alongside the mandatory fields needed for the specific call, the parameters "userName" and "password" must also be populated.

6.3 register.do / registerPreAuth.do – înregistrare comandă

To register an order, use the *register.do / registerPreAuth.do* method. This request is designed for online order registration and automatic redirection to the payment page, followed by the payment confirmation page.

The parameters for registering an order are identical for both types of payments (*1-phase* - by calling *register.do* / *2-phase* - by calling *registerPreAuth.do*).

Based on the URL created with the related parameters (example: username, password, orderNumber, amount, currency, returnUrl, orderBundle[] etc.), the orderId parameter is returned in response, which is created by the system at the time of payment and is unique , as well as a URL in the formUrl parameter.

This parameter (formUrl) contains transaction-specific information, being also the payment page to be presented to the customer (payment page where the card data is entered, followed by the payment confirmation page by entering the static password 12345 - only on the test system).

In case of error, the response will display the errorCode and errorMessage parameters, with the message and error code specific to the problem encountered.

6.3.1 Technical data

Method is accesible only for authenticated users - see chapter 6.2

The parameters required to call the register.do / registerPreAuth.do method:

Name	Type	Mandatory	Description
userName	String(30)	No	Merchant API user - see chapter 6.2
password	String(30)	No	Merchant API password - see chapter 6.2
orderNumber	String(32)	Yes	The unique order number (identifier) in the merchant system. If it is necessary to resume the transaction for any reason, the flow must be resumed with a new orderNumber. We recommend that orderNumber be associated with the invoice number or any other relevant unique ID + iteration number for the same transaction. For example: noInvoice-0, noInvoice-1, noInvoice-2 etc
amount	Decimal(20)	Yes	Order amount in currency subunits (e.g. cents)

Name	Type	Mandatory	Description
currency	Decimal(3)	Yes	Code of the payment currency, according to ISO 4217. <table><tr><td>Name</td><td>Numeric code</td></tr><tr><td>RON</td><td>946</td></tr><tr><td>EUR</td><td>978</td></tr><tr><td>USD</td><td>840</td></tr></table>
returnUrl	String(512)	Yes	The web address to which the customer must be redirected after the payment is completed (both if the payment is successful and if the payment is unsuccessful). This URL must be your finish page and must be in the form http:// or https://
description	String(512)	No	Order description. ASCII characters between 32 and 125 **are allowed** (upper and lower case letters, numbers, the space character and special characters). Diacritics and the ~ character and other non-ASCII characters **are NOT allowed.** Only the first 80 characters will appear on your statement.
language	String(2)	No	Language according to ISO 639-1. If not specified, the system uses the default language in the merchant settings.
pageView	String(20)	No	Possible values: **DESKTOP** – to upload pages designed for displaying on PC monitors **MOBILE** – to upload pages designed for displaying on mobile devices Default value is pageView=DESKTOP
email	String(254)	No	Customer's email address.
childId	String(100)	No (Yes for sub-merchants)	The identifier of the sub-merchant that must belong to the aggregating merchant. The sub-merchant login is used, not the numeric id.
clientId	String(255)	No (Yes for COF)	Customer number (ID) in the merchant system. Available only to merchants with appropriate rights.
bindingId	String(36) - UUID	No (Yes for COF)	The binding identifier that was created previously. It can only be used if the trader has permission to work with bindings. If this parameter is sent in the registerOrder request, the payer will be redirected to a payment page, where only the CVC entry is required.
sessionTimeoutSecs	Decimal(9)	No	Payment page lifetime in seconds.

Name	Type	Mandatory	Description
			! Parameter available only for merchants who have permission agreed with BT. ! If the API call also contains the expirationDate parameter, the sessionTimeoutSecs parameter is ignored.
expirationDate	String(19)	No	Payment page lifetime in format YYYY-MM-DDTHH:mm:ss. ! Parameter available only for merchants who have permission agreed with BT. ! If the API call also contains the sessionTimeoutSecs parameter, the expirationDate parameter takes priority
jsonParams	JSON	NU	Additional field for further information on payment. The format is {"param":"value", "param2":"value2"}
orderBundle (detailed list below)	JSON	Yes	Information on the order details and delivery information about the customer.

* Fields "orderNumber" and its "description" are sent to the bank processing by default (not more than 99 simbols, the following simbols are forbidden to use - %, +, \r, \n).

Format of the **orderBundle** block:

Name	Type	Mandatory	Description
orderCreationDate	String(10)	YES	Date of the order. Date format is YYYY-MM-DD
customerDetails	Object	YES	Details on the customer. See the description of this block contents below.

Format of the **customerDetails** block:

DO NOT use diacritics and the NEWLINE character (ENTER key) in the composition of the message!

Name	Type	Mandatory	Description
email	String(254)	YES	Customer's email address.
phone	String(15)	YES	Customer phone number. It must contain only digits, in international format (eg 40740123123).
contact	String(40)	NO	Contact person.
deliveryInfo	Object	YES	Details on the order delivery that contain parameters described below.
billingInfo	Object	YES	Details on the order delivery that contain parameters described below. Data format is the same as for deliveryInfo[].

Format of the **deliveryInfo** and **billingInfo** block:

DO NOT use diacritics and the NEWLINE character (ENTER key) in the composition of the message!

Name	Tip	Mandatory	Description
deliveryType	String(20)	Mandatory only for **deliveryInfo[]**	Type of the order delivery/billing
country	Decimal(3)	YES	Numeric code for country of delivery/ billing, according to ISO 3166-1
city	String(50)	YES	City of delivery/billing
postAddress	String(50)	YES	Post address of delivery/billing
postAddress2	String(50)	NO	Additional row for address data
postAddress3	String(50)	NO	Additional row for address data
postalCode	String(16)	NO	Postal code of address
state	String(2)	NO	State of address by ISO 3166-2

Please fill in the orderBundle parameter with the real data of the paying customer. If you do not posess enough real data in order to fill the minimum required orderBundle, please do not send this parameter in the payment API

The minimum set of parameters is:

```
{
  "customerDetails":{
    "deliveryInfo":{
      "country":"642",
      "city":"customer_city",
      "postAddress":"customer_address"
    },
    "billingInfo":{
      "country":"642",
      "city":"customer_city ",
      "postAddress":"customer_address"
    }
  }
}
```

Response parameters:

Name	Type	Mandatory	Description
orderId	String(36) – UUID	NO	Unique order number on the payment page. Absent if command registration failed (error is described in errorCode).
formUrl	String(512)	NO	The URL of the payment page to which the customer is redirected after executing the register.do request. An answer returns: • The payment page URL when registering the successful order. • an error if the command registration failed, the error is described in the errorCode field
errorCode	Decimal(3)	NO	The error code that appears during the registration of a payment
errorMessage	String(512)	NO	Error description

Values and description of the errorCode field:

Value	Description
0	No system error.
1	Order number is duplicated, order with given order number is processed already or childId is incorrect
3	Unknown currency
4	Mandatory request parameter was not specified
5	Erroneous value of a request parameter.
7	System error.
8	Orderbundle error.
11	Wrong orderDescription param value

Possible answers for errorMessage:

Value	Description
0	No system error
1	Order with this number was already processed.
1	Order with this number was registered, but was not paid off.
1	Order number is duplicated, order with given order number is processed already
1	Invalid orderNumber
1	Order number is invalid
1	Submerchant could not be found
1	Submerchant is blocked or deleted
3	Unknown currency.
4	Order number is empty
4	Empty merchant user name
4	Empty amount
4	Invalid return URL
4	Empty return URL
4	Password cannot be empty
5	Invalid value of one of the parameters.
5	Wrong value of the language parameter
5	Access denied
5	The user must change his password
5	Invalid [jsonParams]
5	Pre-authorization payment is restricted
7	System error
8	[orderBundle.customerDetails.*] wrong
11	Wrong orderDescription param value

6.3.1.1 returnUrl and the 22 errors

As for the **returnUrl** parameter from register.do / registerPreAuth.do method, it should redirect to the finish page of your site.

In addition to the errors presented in documentation, we also have **22 errors** that you have to treat on your site and which can be obtained using getOrderStatusExtended.do API, the actionCode field. The errors below are the most common and we want to be presented to the customer. For any other error code from the documentation you can present "Transaction refused, please come back ...", or any text that you consider relevant.

Attention! It is forbidden to retry the transaction with the same card for actionCode 803, 804 and 913 ! Advise the client to contact the issuing bank or retry the transaction with another card !

actionCode 104 = Card restricted (temporary or permanent blocks due to no payment or death of cardholder).
actionCode 124 = Transaction cannot be authorized due to government, central bank or financial institution agreement, laws or regulations
actionCode 320 = Inactive card. Please activate the card.
actionCode 801 = Issuer unavailable.
actionCode 803 = Card blocked. Contact the issuing bank or retry the transaction with another card.
actionCode 804 = Transaction not permitted. Contact the issuing bank or retry the transaction with another card.
actionCode 805 = Transaction rejected.
actionCode 861 = Wrong card expiration date.
actionCode 871 = Wrong CVV.
actionCode 905 = Invalid card. It does not exist in the database.
actionCode 906 = Card expired.
actionCode 913 = Invalid transaction. Contact the issuing bank or retry the transaction with another card.
actionCode 914 = Invalid account. Please contact the issuing bank.
actionCode 915 = Insufficient funds.
actionCode 917 = Transaction limit exceeded.
actionCode 952 = Fraud is suspected.
actionCode 998 = Installment transaction is not allowed with this card. Please use a credit card issued by Banca Transilvania.
actionCode 341016 = 3DS2 authentication is declined by Authentication Response (ARes) – issuer
actionCode 341017 = 3DS2 authentication status in ARes is unknown - issuer
actionCode 341018 = 3DS2 CReq cancelled - client
actionCode 341019 = 3DS2 CReq failed - client/issuer
actionCode 341020 = 3DS2 unknown status in RReq – issuer

6.3.2 Examples

6.3.2.1 Payment with register.do

POST Request example:

url: https://ecclients-sandbox.btrl.ro/payment/rest/register.do

body:

userName=*****&password=*****&orderNumber=209123&amount=1200&currency=946&description=testBT&returnUrl=https://magazinulmeu.ro/finish.html&orderBundle={"orderCreationDate":"2020-09-29","customerDetails":{"email":"email@test.com","phone":"40740123456","deliveryInfo":{"deliveryType":"comanda","country":"642","city":"Cluj","postAddress":"Str.Sperantei","postalCode":"12345"},"billingInfo":{"deliveryType":"comanda","country":"642","city":"Cluj","postAddress":"Str.Sperantei","postAddress2":"Str.Sperantei","postAddress3":"Strada","postalCode":"12345"}}}

content type: x-www-form-urlencoded

Response example:

{"orderId":"c4f617e2-edf0-4c23-9408-4d3129f751d8","formUrl":"https://ecclients-sandbox.btrl.ro/payment/merchants/*****/payment.html?mdOrder=c4f617e2-edf0-4c23-9408-4d3129f751d8&language=ro"}

6.3.2.2 Payment with registerPreAuth.do

POST Request example:

url: https://ecclients-sandbox.btrl.ro/payment/rest/registerPreAuth.do

body:

userName=*****&password=*****&orderNumber=8042112&amount=1200&currency=946&description=testBT&returnUrl=https://magazinulmeu.ro/finish.html&orderBundle={"orderCreationDate":"2020-09-29","customerDetails":{"email":"email@test.com","phone":"40740123456","deliveryInfo":{"deliveryType":"comanda","country":"642","city":"Cluj","postAddress":"Str.Sperantei","postalCode":"12345"},"billingInfo":{"deliveryType":"comanda","country":"642","city":"Cluj","postAddress":"Str.Sperantei","postAddress2":"Str.Sperantei","postAddress3":"Strada","postalCode":"12345"}}}

content type: x-www-form-urlencoded

Response example:

{"orderId":"111aa4b9-baeb-4676-bbde-c5e4d5070cce","formUrl":"https://ecclients-sandbox.btrl.ro/payment/merchants/*****/payment.html?mdOrder=111aa4b9-baeb-4676-bbde-c5e4d5070cce&language=ro"}

The formUrl parameter is displayed in the browser and is the payment page provided by Banca Transilvania, being followed by the payment confirmation page by entering the 3DSecure code.

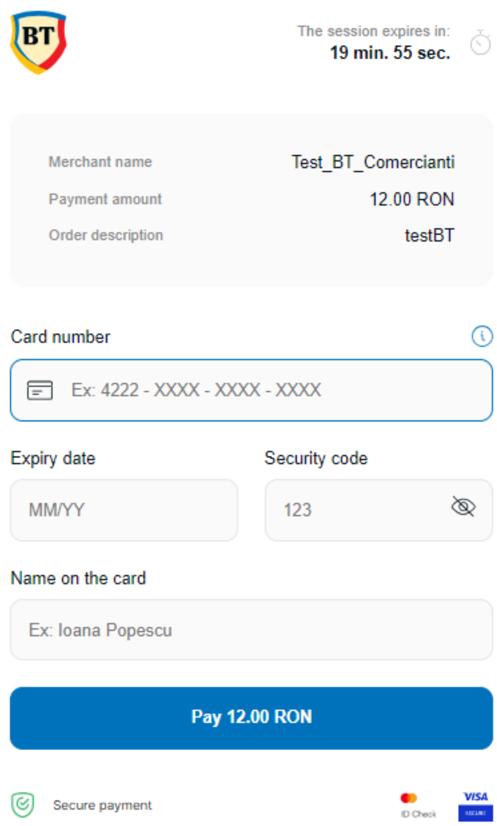

Figure 1. Payment page

As of January 2021, the payment confirmation page is based on two screens. The first screen enters the dynamic password received via SMS, and the second screen enters the unique password defined by the cardholder.

HERE is the web page that contains information about confirming online payments.

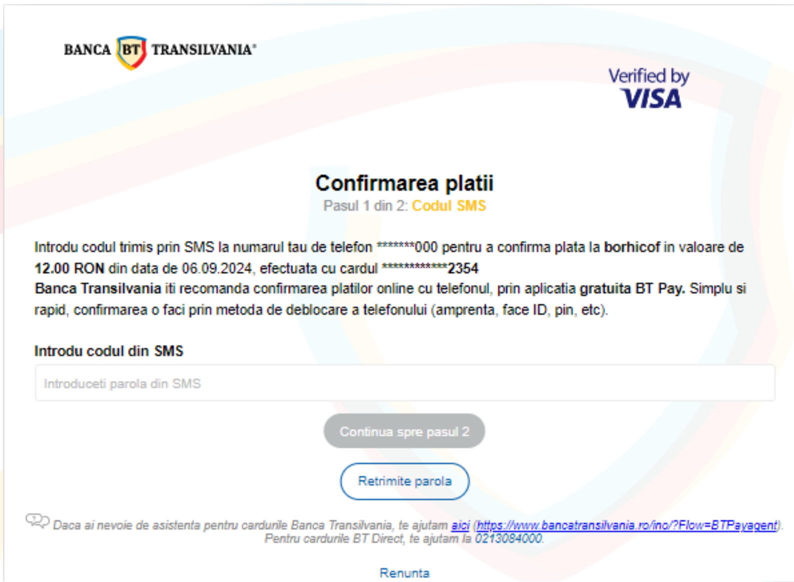

Figure 2. Payment confirmation page with 3DS2 via SMS code

Figure 3. 3DS2 payment confirmation page with a single password

Because the return page of the BT platform is entered in the returnUrl parameter, the client is redirected to the page below and contains information on the order number, approval code, terminal ID, reference number, amount, card details (hidden pan, expiration date and cardholder's name), as well as the description provided by the merchant in the description field.

In your call, the **returnUrl** parameter must contain the address / link of your finish page in which you present the transaction status to the customer based on the response you receive after calling the **getOrderStatusExtended.do** API and in which the most common errors are treated (found in the "*returnUrl and the 22 errors*" section).

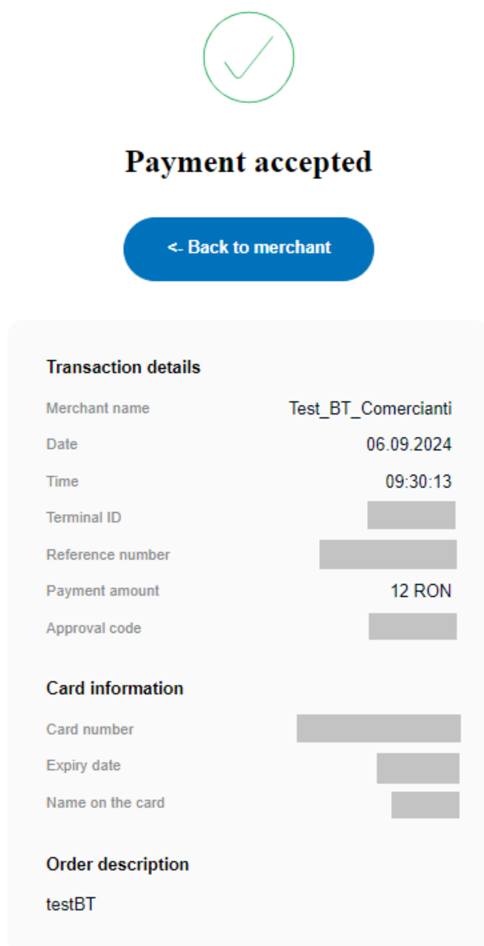

Figure 4. Finish payment page

6.3.2.3 Payment register.do declined

POST Request example:

url: https://ecclients-sandbox.btrl.ro/payment/rest/register.do

body:

userName=*****&password=*****&orderNumber=209126&amount=1000&currency=946&description=testBT&returnUrl=https://magazinulmeu.ro/finish.html&orderBundle={"orderCreationDate":"2020-09-29","customerDetails":{"email":"email@test.com","phone":"40740123456","deliveryInfo":{"deliveryType":"comanda","country":"642","city":"Cluj","postAddress":"Str.Sperantei","postalCode":"12345"},"billingInfo":{"deliveryType":"comanda","country":"642","city":"Cluj","postAddress":"Str.Sperantei","postAddress2":"Str.Sperantei","postAddress3":"Strada","postalCode":"12345"}}}

content type: x-www-form-urlencoded

Response example:

{"orderId":"c6b2ea18-a0e0-4a69-8666-1dc1d60b6517","formUrl":"https://ecclients-sandbox.btrl.ro/payment/merchants/*****/payment.html?mdOrder=c6b2ea18-a0e0-4a69-8666-1dc1d60b6517&language=ro"}

6.3.2.4 Payment registerPreAuth.do declined

POST Request example:

url: https://ecclients-sandbox.btrl.ro/payment/rest/registerPreAuth.do

body:

userName=*****&password=*****&orderNumber=8042114&amount=1000&currency=946&description=testBT&returnUrl=https://magazinulmeu.ro/finish.html&orderBundle={"orderCreationDate":"2020-09-29","customerDetails":{"email":"email@test.com","phone":"40740123456","deliveryInfo":{"deliveryType":"comanda","country":"642","city":"Cluj","postAddress":"Str.Sperantei","postalCode":"12345"},"billingInfo":{"deliveryType":"comanda","country":"642","city":"Cluj","postAddress":"Str.Sperantei","postAddress2":"Str.Sperantei","postAddress3":"Strada","postalCode":"12345"}}}

content type: x-www-form-urlencoded

Response example:

{"orderId":"9c3ee654-d8a9-4620-b29e-9698ad4962da","formUrl":"https://ecclients-sandbox.btrl.ro/payment/merchants/*****/payment.html?mdOrder=9c3ee654-d8a9-4620-b29e-9698ad4962da&language=ro"}

In the payment page we enter the wrong expiration date. The finish page returns the answer in the image below:

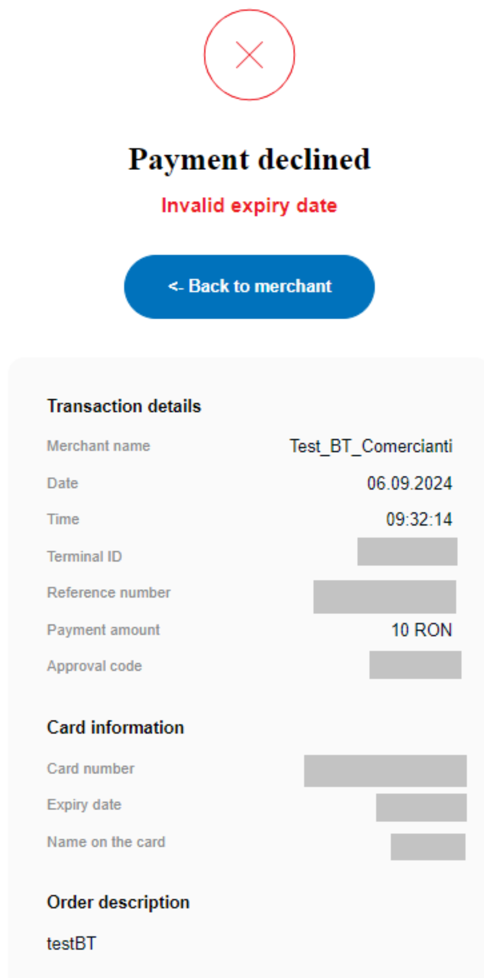

Figure 5. Payment declined

Our request is that the finish page be on your site and in which to treat the most common errors specified in the "*returnUrl and the 22 errors*" section.

6.4 deposit.do - cash payments - for 2-phase payments

For payments made with 2-phase, after the payment is made by the customer, the successful transactions are with **APPROVED** status, which means that the customer's money for the order placed is blocked.

Upon delivery / confirmation of delivery of goods is made "completion" of the transaction, collection of the amount related to the transaction, being necessary to implement the deposit.do method. This operation can only be performed once.

If this operation is NOT performed, you will NOT cash the transaction.

The collection of transactions is only possible for transactions with **APPROVED** status and can be done either for the entire pre-authorized amount or for a lower amount. If the collection operation is performed for an amount lower than the pre-authorized amount, the difference is automatically unlocked on the customer's account.

In this phase, the status of the initial transaction, the one made by the customer at the time of placing the order, changes from **APPROVED to DEPOSITED** for those transactions that did not encounter any errors when placing the order. If the initial transaction is not with “Approved” status, the amount related to the transaction cannot be unlocked.

6.4.1 Technical data

Method is accesible only for authenticated users - see chapter 6.2.

The parameters required to call the deposit.do method:

Name	Type	Mandatory	Description
userName	String(30)	No	Merchant API user - see chapter 6.2
password	String(30)	No	Merchant API password - see chapter 6.2
orderId	String(36)-UUID	YES	Unique order number on the payment page
amount	Decimal(20)	YES	Order amount in currency subunits (e.g. cents)
depositLoyalty	boolean	No	For combined transactions (RON + LOY). By default, the parameter is set to false. If the parameter is true, the collection operation is performed for both transactions, using the orderId of the RON transaction. Attention to the amount entered in the amount parameter: for total collection, this must be the sum between the transaction in RON and LOY. Possible values: true / false.

Response parameters:

Name	Type	Mandatory	Description
errorCode	Decimal(3)	No	Error code
errorMessage	String(512)	No	Error description
actionCode	Decimal(10)	YES	Authorization code provided by the processing system
actionCodeDescription	String(512)	YES	Description of the code provided by the actionCode parameter
loyaltyOperationActionCode	Decimal(10)	No	Authorization code provided by the processing system for the loy transaction
loyaltyOperationActionCodeDescription	String(512)	No	Description of the code provided by the loyaltyOperationActionCode parameter

Values and description of the errorCode field:

Value	Description
0	No system error
5	Erroneous value of a request parameter
6	Unregistered OrderId
7	System error or operation not possible for the transaction status
8	Collection exceeds pre-authorized amount

Possible answers for errorMessage:

Value	Description
0	Success
5	Access denied
5	The user must change his password
5	Invalid amount
5	Deposit amount must be zero, or more than 1 currency unit (e.g. 1 euro)
5	deposit amount is invalid
6	[orderId] is empty
6	No such order
7	Payment must be in approved state
7	Deposit is impossible chargeback flag is present.
7	System error
8	deposited amount is greater then registered amount

6.4.2 Examples

6.4.2.1 Successful collection

The initial transaction for the example below is presented in the "Payment registerPreAuth.do" section.

POST Request example:

url: https://ecclients-sandbox.btrl.ro/payment/rest/deposit.do

body:

userName=*****&password=*****&orderId=111aa4b9-baeb-4676-bbde-c5e4d5070cce&amount=1200

content type: x-www-form-urlencoded

Exemplu response

{"errorCode":"0","errorMessage":"Success","actionCode":0,"actionCodeDescription":"Payment approved and completed successfully."}

6.4.2.2 Declined collection

The initial transaction for the example below is presented in the "RegisterPreAuth.do declined payment" section.

POST Request example:

url: https://ecclients-sandbox.btrl.ro/payment/rest/deposit.do

body:

userName=*****&password=*****&orderId=9c3ee654-d8a9-4620-b29e-9698ad4962da&amount=950

content type: x-www-form-urlencoded

Response example:

{"errorCode":"7","errorMessage":"Payment must be in approved state"}

6.4.2.3 Successful collection of split transaction (RON + LOY)

Transaction with the amount of 30 RON (amount=3000), paid by the customer in:

- 12 RON – orderId 4b825a21-7c3c-4249-ad48-aff3efc268e2

- 18 LOY – orderId 5cf834d4-fdcc-4b16-b980-1673b4ccc66a

POST Request example:

url: https://ecclients-sandbox.btrl.ro/payment/rest/deposit.do

body:

userName=*****&password=*****&orderId=4b825a21-7c3c-4249-ad48-aff3efc268e2&amount=3000&depositLoyalty=true

content type: x-www-form-urlencoded

Response example:

{"errorCode":"0","errorMessage":"Success","actionCode":0,"actionCodeDescription":"Payment approved and completed successfully", "loyaltyOperationActionCode":0,"loyaltyOperationActionCodeDescription":"Payment approved and completed successfully"}

6.4.2.4 Declined collection for split transaction (RON + LOY)

POST Request example:

url: https://ecclients-sandbox.btrl.ro/payment/rest/deposit.do

body:

userName=*****&password=*****&orderId=4b825a21-7c3c-4249-ad48-aff3efc268e2&amount=4000&depositLoyalty=true

content type: x-www-form-urlencoded

Response example:

{"errorCode":"7","errorMessage":"Payment must be in approved state"}

6.5 reverse.do – pre-authorization cancellation - for 2-phase payments

In the case of payments made with 2-phase, the cancellation of the pre-authorization is available for orders with **APPROVED** status (the status of the order changes from "Approved" to "Reversed").

The operation can be performed only once. If the undo operation caused an error, the next attempt will not be successful.

The reversal operation is only active for traders with appropriate rights (in agreement with the bank) and is available for a limited period of time.

If you do not honor the order, please complete the reverse operation within a **maximum of 24 hours from the date of the transaction.**

6.5.1 Technical data

Method is accesible only for authenticated users - see chapter 6.2.

The parameters required to call the reverse.do method:

Name	Type	Mandatory	Description
userName	String(30)	No	Merchant API user - see chapter 6.2
password	String(30)	No	Merchant API password - see chapter 6.2
orderId	String(36)-UUID	YES	Unique order number on the payment page
reverseLoyalty	boolean	No	For combined transactions (RON + LOY). By default, the parameter is set to false. If the parameter is true, the reversal operation is performed for both transactions, using the orderId of the RON transaction. Possible values: true / false.

Response parameters:

Name	Type	Mandatory	Description
errorCode	Decimal(3)	NO	Error code
errorMessage	String(512)	NO	Error description
actionCode	Decimal(10)	YES	Authorization code provided by the processing system
loyaltyOperationActionCode	Decimal(10)	NO	Authorization code provided by the processing system for the loy transaction

Values and description of the errorCode field

Value	Description
0	No system error
5	Erroneous value of a request parameter
6	Unregistered OrderId
7	System error

Possible answers for errorMessage:

Value	Description
0	Success
5	Access denied
5	System error
5	The user must change his password
5	[orderId] is empty
6	No such order
7	Payment must be in a correct state
7	Reversal is impossible. Reason: wrong internal values, check hold and deposited amounts
7	Reversal is impossible for current transaction state chargeback flag is present.
7	System error

6.5.2 Examples

In order to be able to exemplify this method, it is necessary that the initial transaction has an "Approved" status. Because the examples presented above are with "Deposited" status as a result of unlocking the amount, I will perform two more transactions, one successful and one declined due to the wrong expiration date, using the registerPreAuth.do method.

6.5.2.1 Successful cancellation

The example is for the successful transaction and "Approved" status.

OrderId: 26925da2-9c33-46db-81b0-ba7e00ee3a83	OrderNumber: 8042117
Amount: 650	Currency: 946

POST Request example:

url: https://ecclients-sandbox.btrl.ro/payment/rest/reverse.do

body:

userName=*****&password=******&orderId=26925da2-9c33-46db-81b0-ba7e00ee3a83

content type: x-www-form-urlencoded

Response example:

{"errorCode":"0","errorMessage":"Success","actionCode":0}

6.5.2.2 Cancellation declined

The example is for the declined transaction and “Declined” status.

OrderId: 6ac8a1b5-21a5-4b63-a62b-04285039da8e	OrderNumber: 8042118
Amount: 570	Currency: 946

POST Request example:

url: https://ecclients-sandbox.btrl.ro/payment/rest/reverse.do

body:

userName=*****&password=******&orderId=6ac8a1b5-21a5-4b63-a62b-04285039da8e

content type: x-www-form-urlencoded

Response example:

{"errorCode":"5","errorMessage":"Access denied"}

6.5.2.3 Successful cancellation of split transaction (RON + LOY)

OrderId RON: 7956a9b9-29d8-4964-bbb2-33264414347e
OrderID LOY: 6676628a-a5e0-4476-83b0-3921fd2998d7

POST Request example:

url: https://ecclients-sandbox.btrl.ro/payment/rest/reverse.do

body:

userName=*****&password=******&orderId=7956a9b9-29d8-4964-bbb2-33264414347e&reverseLoyalty=true

content type: x-www-form-urlencoded

Exemplu response:

{"errorCode":"0","errorMessage":"Success","actionCode":0,
"loyaltyOperationActionCode":0}

6.6 refund.do - repayment

If the customer leaves the service (in case of payments made through 1-phase) or returns one or more products (in case of payments made through 2-phase), iPay allows the partial or total refund of the money for the respective order back to the customer's account, by implementing the refund.do method.

To perform the refund operation, the status of the order must be **DEPOSITED or PARTIALLY REFUNDED.**

Multiple refunds are allowed, but their total amount cannot exceed the amount paid by the customer for the order placed.

6.6.1 Technical data

Method is accesible only for authenticated users - see chapter 6.2.

The parameters required to call the refund.do method:

Name	Type	Mandatory	Description
userName	String(30)	No	Merchant API user - see chapter 6.2
password	String(30)	No	Merchant API password - see chapter 6.2
orderId	String(36)-UUID	Yes	Unique order number on the payment page
amount	Decimal(20)	Yes	Order amount in currency subunits (e.g. cents)
refundLoyalty	boolean	No	For combined transactions (RON + LOY). By default, the parameter is set to false. If the parameter is true, the refund operation is performed for both transactions, using the orderId of the RON transaction. Attention to the amount entered in the amount parameter: for total refund, this must be the sum between the transaction in RON and LOY. Possible values: true / false.

Response parameters:

Name	Type	Mandatory	Description
errorCode	Decimal(3)	No	Error code
errorMessage	String(512)	No	Description of the error
actionCode	Decimal(10)	Yes	Processing system authorization code
actionCodeDescription	String(512)	YES	Description of the code provided by the actionCode parameter

Name	Type	Mandatory	Description
loyaltyOperationActionCode	Decimal(10)	No	Authorization code provided by the processing system for the loy transaction
loyaltyOperationActionCodeDescription	String(512)	No	Description of the code provided by the loyaltyOperationActionCode parameter

Values and description of the errorCode field:

Value	Description
0	No system error
5	Erroneous value of a request parameter
6	Unregistered OrderId
7	System error
8	The refunded amount exceeds the collected amount

Possible answers for errorMessage:

Value	Description
0	Success
5	Access denied
5	System error
5	The user must change his password
5	[orderId] is empty
5	Amount is invalid
6	No such order
7	Refund is impossible for current transaction state
7	Wrong deposit amount (less than 1 currency unit, e.g. 1 euro).
7	Refund is impossible chargeback flag is present.
7	System error.
8	refund amount exceeded

6.6.2 Examples

6.6.2.1 Successful refund - 1-phase

We want to refund the amount of 12 RON from the initial transaction in the amount of 12 RON exemplified in the "Payment register.do" section.

POST Request example:

url: https://ecclients-sandbox.btrl.ro/payment/rest/refund.do

body:

userName=*****&password=*****&orderId= c4f617e2-edf0-4c23-9408-4d3129f751d8&amount=100

content type: x-www-form-urlencoded

Response example:

{"errorCode":"0","errorMessage":"Success","actionCode":0,"actionCodeDescription":"actionCode000"}

6.6.2.2 Successful refund - 2-phase

We want to refund the amount of 3 RON from the initial transaction in the amount of 12 RON exemplified in the "Payment registerPreAuth.do" section.

Attention! The refund operation for 2-phase payments can only be performed after the collection operation (deposit.do).

POST Request example:

url: https://ecclients-sandbox.btrl.ro/payment/rest/refund.do

body:

userName=*****&password=******&orderId=111aa4b9-baeb-4676-bbde-c5e4d5070cce&amount=300

content type: x-www-form-urlencoded

Response example:

{"errorCode":"0","errorMessage":"Success","actionCode":0,"actionCodeDescription":"actionCode000"}

6.6.2.3 Refund declined

We want to refund 4 RON from the initial transaction in the amount of 10 RON exemplified in the “Register.do payment declined” section. Because this transaction was declined due to "Wrong card expiration date.", Refund is not possible.

POST Request example:

url: https://ecclients-sandbox.btrl.ro/payment/rest/refund.do

body:

userName=*****&password=*****&orderId=c6b2ea18-a0e0-4a69-8666-1dc1d60b6517&amount=400

content type: x-www-form-urlencoded

Response example:

{"errorCode":"7","errorMessage":"Refund is impossible for current transaction state"}

6.6.2.4 Successful refund for split transaction (RON + LOY)

Refund operation for transaction paid by customer with amount 40 RON (amount=4000):

- 26 RON – orderId d5e7e37c-5a79-4262-addc-78b87cadbdf5
- 14 LOY – orderId e05c5c61-40ae-4ec3-971a-5a3858983362

POST Request example:

url: https://ecclients-sandbox.btrl.ro/payment/rest/refund.do

body:

userName=*****&password=*****&orderId=d5e7e37c-5a79-4262-addc-78b87cadbdf5&amount=4000&refundLoyalty=true

content type: x-www-form-urlencoded

Response example:

{"errorCode":"0","errorMessage":"Success","actionCode":0,"actionCodeDescription":"Payment approved and completed successfully", "loyaltyOperationActionCode":0,"loyaltyOperationActionCodeDescription":"Payment approved and completed successfully"}

6.7 getOrderStatusExtended.do

Use the getOrderStatusExtended.do method to display on the finish page the answer about the transaction (approved or declined and the reason for the decline of the transaction), as well as other information about the transaction of a registered order.

6.7.1 Technical data

Method is accesible only for authenticated users - see chapter 6.2.

The parameters required to call the getOrderStatusExtended.do method:

Name	Type	Mandatory	Description
userName	String(30)	No	Merchant API user - see chapter 6.2
password	String(30)	No	Merchant API password - see chapter 6.2
orderId	String(36)-UUID	yes*	Unique order number on the payment page
orderNumber	String(32)	yes*	The unique order number (identifier) in the merchant system. If it is necessary to resume the transaction for any reason, the flow must be resumed with a new orderNumber. We recommend that orderNumber be associated with the invoice number or any other relevant unique ID + iteration number for the same transaction. For example: noInvoice-0, noInvoice-1, noInvoice-2 etc
includeCardArt	boolean	No	Control of displaying the information from bindingInfo for the cardArtPicture and cardArtForegroundColor parameters. Possible values: true / false. Default value: false.

* It is necessary to specify whether orderId or orderNumber in the request. If the request contains both parameters, orderId is a priority.

Response parameters:

Name	Type	Mandatory	Description
errorCode	Decimal(3)	No	Error code
errorMessage	String(512)	No	Error description
orderNumber	String(32)	Yes	Number (identifier) of the order in the merchant's system
orderStatus	Decimal(2)	No	Order status in the payment gate. The value is selected from the variants listed

<table>
<tr><th>Name</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>below. Absent, if no matching order was found.</td></tr>
<tr><td>actionCode</td><td>Decimal(10)</td><td>Yes</td><td>Processing system authorization code</td></tr>
<tr><td>actionCodeDescription</td><td>String(512)</td><td>Yes</td><td>Description of the code provided by the actionCode parameter</td></tr>
<tr><td>paymentWay</td><td>String(32)</td><td>No</td><td>Payment method (CARD / CARD_BINDING / BT_PAY)</td></tr>
<tr><td>amount</td><td>Decimal(20)</td><td>Yes</td><td>Order amount in currency subunits (e.g. cents)</td></tr>
<tr><td>currency</td><td>Decimal(3)</td><td>No</td><td>Code of the payment currency, according to ISO 4217.<table><tr><td>Name</td><td>Numeric code</td></tr><tr><td>RON</td><td>946</td></tr><tr><td>EUR</td><td>978</td></tr><tr><td>USD</td><td>840</td></tr></table></td></tr>
<tr><td>date</td><td>Decimal(20)</td><td>Yes</td><td>Order registration date in milliseconds past Unix Epoch (January 1, 1970 UTC)</td></tr>
<tr><td>orderDescription</td><td>String(512)</td><td>No</td><td>Free-formed description of the order</td></tr>
<tr><td>ip</td><td>String(15) – IPv4</td><td>Yes</td><td>IP address of the user who payed for the order</td></tr>
<tr><td colspan="4">merchantOrderParams[] (element includes name and value)</td></tr>
<tr><td>name</td><td>String(20)</td><td>Yes</td><td>Name of the merchant additional parameter. For payments paid in RON + starBT points, "loyaltyAmount" will be displayed</td></tr>
<tr><td>value</td><td>String(1024)</td><td>Yes</td><td>Value of the merchant additional parameter. For payments paid in RON + starBT points, the amount of money paid in loyalty points will be displayed.</td></tr>
<tr><td colspan="4">attributes[] (element includes name and value)</td></tr>
<tr><td>name</td><td>String(20)</td><td>Yes</td><td>Attribute name – "mdOrder", "loyalties", "installment"</td></tr>
<tr><td>value</td><td>String(1024)</td><td>Yes</td><td>The value of the attribute e.g. for:
- "mdOrder" - order identifier in the payment system (unique in the system) - UUID format
- "loyalties" - mdOrder and orderNumber of the paid transaction in loyalty points will be displayed.
- "installment" - the number of installments</td></tr>
</table>

Name	Type	Mandatory	Description
cardAuthInfo (includes the parameters pan, expiration, cardholderName and approvalCode and the list of secureAuthInfo elements):			
pan	String(23)	No	Masked number of the card that was used in payment. Specified only for paid orders.
expiration	Decimal(6)	No	Card expiration date in the YYYYMM format. Specified only for paid orders.
cardholderName	String(64)	No	Cardholder name. Specified only for paid orders.
approvalCode	String(6)	No	IPS authorization code
secureAuthInfo (element includes eci element and elements of threeDSInfo type, which is a list of cavv and xid):			
eci	Decimal(4)	No	Electronic Commerce Indicator
cavv	String(200)	No	Cardholder Authentication Verification Value
xid	String(80)	No	Electronic Commerce Transaction Identifier
bindingInfo (element consits of clientId, bindingId, cardArtUrl, cardArtPicture, cardArtForegroundColor, bin and panLastFour):			
clientId	String(255)	No (Yes for COF)	Client number (ID) in the shop system, transferred during the order registration. Presents only if a shop is allowed to create links.
bindingId	String(36) - UUID	No (Yes for COF)	The identifier of a link created during order payment or used for payment. Presents only if a shop is allowed to create links.
cardArtUrl	String	No (for COF)	For MasterCard cards. The URL with the image of the respective card (only if the card picture is available).
cardArtPicture	String	No (for COF)	For Visa cards. Image of the respective card in png base64 format (only if the card picture is available).
cardArtForegroundColor	String(6)	No (for COF)	Text color in RGB format.
bin	String(6)	No (for COF)	The first 6 digits of the physical card associated with the token.
panLastFour	String(4)	No (for COF)	The last 4 digits of the physical card associated with the token.
authDateTime	Decimal(20)	No	Authorization date and time in milliseconds past Unix Epoch (January 1, 1970 UTC)
authRefNum	String(24)	No	The transaction identifier in processing system - reference number - RRN

Name	Type	Mandatory	Description
terminalId	String(10)	No	Terminal Id
paymentAmountInfo element (consists of approvedAmount, depositedAmount, refundedAmount and paymentState parameters):			
approvedAmount	Decimal(20)	No	The amount withheld on the customer's card
depositedAmount	Decimal(20)	No	Amount confirmed for deposit
refundedAmount	Decimal(20)	No	Refund amount
paymentState	String(9)	No	Payment state
bankInfo element (consist of bankName, bankCountryCode and bankCountryName parameters):			
bankName	String(200)	No	Name of issuing bank
bankCountryCode	String(4)	No	Code of issuing bank country
bankCountryName	String(160)	No	The country of the issuing bank
orderBundle (detailed list below)	JSON	YES	Information on the order details and delivery information about the customer.
chargeback	boolean	NO	Indication if a transaction marked as chargeback
refunds[] (includes referenceNumber, actionCode, amount and date):			
referenceNumber	String(100)	NO	The transaction identifier in processing system - reference number - RRN
actionCode	String(3)	NO	Authorization code provided by the processing system
amount	Decimal(20)	NO	Refunded amount displayed in currency subunits (e.g. cents)
date	String(100)	NO	Date and time of processing the refund operation

The detail of the **orderBundle[]** field is:

Name	Type	Mandatory	Description
orderCreationDate	Decimal(13)	Yes	Order date in Unix Time format
customerDetails	Object	Yes	Details on the customer. See the description of this block contents below.

Format of the **customerDetails[]** block:

Name	Type	Mandatory	Description
email	String(254)	Yes	Customer's email address.
phone	Decimal(15)	Yes	Customer phone number. It must contain only digits, in international format (eg 40740123123).
contact	String(40)	No	Contact person.
deliveryInfo	Object	Yes	Details on the order delivery that contain parameters described below.
billingInfo	Object	Yes	Details on the order delivery that contain parameters described below. Data format is the same as for deliveryInfo[].

Format of the **deliveryInfo** and **billingInfo** block:

Name	Type	Mandatory	Description
deliveryType	String(20)	Mandatory only for **deliveryInfo[]**	Type of the order delivery/billing
country	Decimal(3)	Yes	Numeric code for country of delivery/ billing, according to ISO 3166-1
city	String(50)	Yes	City of delivery/billing
postAddress	String(50)	Yes	Post address of delivery/billing
postAddress2	String(50)	No	Additional row for address data
postAddress3	String(50)	No	Additional row for address data
postalCode	String(16)	No	Postal code of address
state	String(2)	No	State of address by ISO-3166-2

Values and description of the errorCode field:

Value	Description
0	No system error
1	Expected [orderId] или [orderNumber]
2	The order is declined because of an error in the payment credentials
5	Access denied
5	The user must change his password
6	Unregistered OrderId
7	System error

The orderStatus field may have the following statuses:

State number	Description
0	Order registered, but not paid off
1	Pre-authorisation amount was held (for two-phase payment)
2	The amount was deposited successfully
3	Authorization reversed
4	Transaction was fully refunded
5	Authorization through the issuer's ACS initiated.
6	Authorization declined
7	Transaction was partially refunded

6.7.2 Examples

6.7.2.1 getOrderStatusExtended for 1-phase accepted transaction

The information in the response corresponds to the transaction made in the "Payment register.do" section.

POST Request example:

url: https://ecclients-sandbox.btrl.ro/payment/rest/getOrderStatusExtended.do

body:

userName=*****&password=******&orderId=c4f617e2-edf0-4c23-9408-4d3129f751d8

content type: x-www-form-urlencoded

Response example:

{"errorCode":"0","errorMessage":"Success","orderNumber":"209123",**"orderStatus":4,"actionCode":0,"actionCodeDescription":"Request processed successfully"**,"amount":1200,"currency":"946","date":1601369648420,"orderDescription":"test BT","ip":"***.***.**.**","merchantOrderParams":[{"name":"numberTime","value":"21.041999999999998"},{"name":"paymentTime","value":"196.472"}],"attributes":[{"name":"mdOrder","value":"c4f617e2-edf0-4c23-9408-4d3129f751d8"}],"cardAuthInfo":{"expiration":"******","cardholderName":"test","approvalCode":"687541","pan":"********"},"authDateTime":1601370323967,"terminalId":"********","authRefNum":"002311094120","paymentAmountInfo":{"paymentState":"REFUNDED","approvedAmount":1200,"depositedAmount":0,"refundedAmount":1200},"bankInfo":{"bankName":"BANCA TRANSILVANIA S.A.","bankCountryCode":"RO","bankCountryName":"Romania"},"orderBundle":{"orderCreationDate":1601337600000,"customerDetails":{"email":"email@test.com","phone":"40740123456","deliveryInfo":{"deliveryType":"comanda","country":"642","city":"Cluj","postAddress":"Str.Sperantei","postalCode":"12345"},"billingInfo":{"country":"642","city":"Cluj","postAddress":"Str.Sperantei","postAddress2":"Str.Sperantei","postAddress3":"Strada","postalCode":"12345"}}}}

6.7.2.2 getOrderStatusExtended for 2-phase accepted transaction

The information in the response corresponds to the transaction made in the "Payment registerPreAuth.do" section.

POST Request example:

url: https://ecclients-sandbox.btrl.ro/payment/rest/getOrderStatusExtended.do

body:

userName=*****&password=******&orderId=111aa4b9-baeb-4676-bbde-c5e4d5070cce

content type: x-www-form-urlencoded

Response example:

{"errorCode":"0","errorMessage":"Success","orderNumber":"8042112",**"orderStatus":2,"actionCode":0,"actionCodeDescription":"Request processed successfully"**,"amount":1200,"currency":"946","date":1601382408093,"orderDescription":"test BT","ip":"***.***.**.**","merchantOrderParams":[{"name":"paymentTime","value":"56.786"},{"name":"numberTime","value":"10.94"}],"attributes":[{"name":"mdOrder","value":"111aa4b9-baeb-4676-bbde-c5e4d5070cce"}],"cardAuthInfo":{"expiration":"******","cardholderName":"test","approvalCode":"627744","pan":"********"},"authDateTime":1601382823134,"terminalId":"********","authRefNum":"002311627744","paymentAmountInfo":{"paymentState":"DEPOSITED","approvedAmount":1200,"depositedAmount":1200,"refundedAmount":0},"bankInfo":{"bankName":"BANCA TRANSILVANIA S.A.","bankCountryCode":"RO","bankCountryName":"Romania"},"orderBundle":{"orderCreationDate":1601337600000,"customerDetails":{"email":"email@test.com","phone":"40740123456","deliveryInfo":{"deliveryType":"comanda","country":"642","city":"Cluj","postAddress":"Str.Sperantei","postalCode":"12345"},"billingInfo":{"country":"642","city":"Cluj","postAddress":"Str.Sperantei","postAddress2":"Str.Sperantei","postAddress3":"Strada","postalCode":"12345"}}}}

6.7.2.3 getOrderStatusExtended for declined transaction

The information in the response corresponds to the transaction made in the "Payment register.do declined" section.

POST Request example:

url: https://ecclients-sandbox.btrl.ro/payment/rest/getOrderStatusExtended.do

body:

userName=*****&password=******&orderId=c6b2ea18-a0e0-4a69-8666-1dc1d60b6517

content type: x-www-form-urlencoded

Response example:

{"errorCode":"0","errorMessage":"Success","orderNumber":"209126",**"orderStatus":6,"actionCode":861,"actionCodeDescription":"Invalid expiry date."**,"amount":1000,"currency":"946","date":1601371605534,"orderDescription":"testBT","ip":"***.***.**.**","merchantOrderParams":[{"name":"paymentTime","value":"28.652"},{"name":"numberTime","value":"4.066"}],"attributes":[{"name":"mdOrder","value":"c6b2ea18-a0e0-4a69-8666-1dc1d60b6517"}],"cardAuthInfo":{"expiration":"******","cardholderName":"tranz declinata","approvalCode":"000000","pan":"********"},"terminalId":"********","authRefNum":"002311095612","paymentAmountInfo":{"paymentState":"DECLINED","approvedAmount":0,"depositedAmount":0,"refundedAmount":0},"bankInfo":{"bankName":"BANCA TRANSILVANIA S.A.","bankCountryCode":"RO","bankCountryName":"Romania"},"orderBundle":{"orderCreationDate":1601337600000,"customerDetails":{"email":"email@test.com","phone":"40740123456","deliveryInfo":{"deliveryType":"comanda","country":"642","city":"Cluj","postAddress":"Str.Sperantei","postalCode":"12345"},"billingInfo":{"country":"642","city":"Cluj","postAddress":"Str.Sperantei","postAddress2":"Str.Sperantei","postAddress3":"Strada","postalCode":"12345"}}}}

6.7.2.1 getOrderStatusExtended for split transaction (RON + LOY)

POST Request example:

url: https://ecclients-sandbox.btrl.ro/payment/rest/getOrderStatusExtended.do

body:

userName=*****&password=******&orderId=d5e7e37c-5a79-4262-addc-78b87cadbdf5

content type: x-www-form-urlencoded

Response example:

{"errorCode":"0","errorMessage":"Success","orderNumber":"2026021818322324ew","orderStatus":4,"actionCode":0,"actionCodeDescription":"Request processed successfully","paymentWay":"CARD","amount":2600,"currency":"946","date":1771435057969,"orderDescription":"Comanda de la comerciantul Test Cris este:","ip":"***.***.**.**","merchantOrderParams":[{"name":"discountedAmount"},{"name":"FORCE_3DS2","value":"true"},{"name":"numberTime","value":"0.666"},{"name":"paymentTime","value":"12.608"},{"name":"loyaltyAmount","value":"1400"}],"attributes":[{"name":"mdOrder","value":"d5e7e37c-5a79-4262-addc-78b87cadbdf5"},{"name":"loyalties","value":"[{mdOrder:e05c5c61-40ae-4ec3-971a-5a3858983362,orderNumber:1771435076398}]"}],"cardAuthInfo":{"expiration":"******","cardholderName":"test split","approvalCode":"151561","pan":"********"},"authDateTime":1771435075354,"terminalId":"********","authRefNum":"230177409921","paymentAmountInfo":{"paymentState":"REFUNDED","approvedAmount":2600,"depositedAmount":0,"refundedAmount":2600},"bankInfo":{"bankName":"BANCA TRANSILVANIA S.A.","bankCountryCode":"RO","bankCountryName":"Romania"},"orderBundle":{"orderCreationDate":1557619200000,"customerDetails":{"email":"email@test.com","phone":"4073119839","contact":"contact_text","deliveryInfo":{"deliveryType":"thistype","country":"642","city":"Cluj","postAddress":"Strada Speraneti nr 132","postalCode":"12345"},"billingInfo":{"country":"642","city":"Cluj","postAddress":"Strada Speraneti nr 132","postAddress2":"Strada Speraneti nr 132","postAddress3":"Strada Speraneti nr 132","postalCode":"12345"}}},"refunds":[{"referenceNumber":"230177409921","actionCode":"0","amount":2600,"date":"20260218191929"}],"chargeback":false}

6.8 getFinishedPaymentInfo.do

Use the getFinishedPaymentInfo.do method to display on the finish page the response to the transaction (approved or declined and the reason for declining the transaction), as well as other information about the transaction of a registered order - for payments with a payment link from the iPAY console.

The recommended method is POST with content-type x-www-form-urlencoded and the parameters set in the request body.

API for:

- Test environment - https://ecclients-sandbox.btrl.ro/payment/rest/getFinishedPaymentInfo.do + mandatory parameters
- Production environment - https://ecclients.btrl.ro/payment/rest/getFinishedPaymentInfo.do + mandatory parameters

6.8.1 Tehnical data

This endpoint does not require authentication. It will not contain the authentication header, nor the parameters "userName" and "password" in the body of the call.

The parameters required to call the getFinishedPaymentInfo.do method:

Name	Type	Mandatory	Description
orderId	String(36)-UUID	yes	Unique order number on the payment page
token	String(512)	yes	Internally generated ID that is linked to the payment page. It is a temporary token, with a validity of 10 minutes, through which the result of the order can be obtained.
language	String(2)	no	Language according to ISO 639-1. If not specified, the system uses the default language in the merchant settings.

Response parameters:

Name	Type	Mandatory	Description
actionCode	Decimal(10)	No	Processing system authorization code
actionDesc	String(512)	No	Description of the code provided by the actionCode parameter
mdOrder	String(36)-UUID	No	Unique order number on the payment page
orderNumber	String(32)	No	Unique order number
orderDescription	String(512)	No	Free-formed description of the order

Name	Type	Mandatory	Description
amountFormatted	Float	No	Formatted order amount
currencyName	String(3)	No	Name currency
paymentRefNum	String(100)	No	The transaction identifier in processing system - reference number - RRN
approvalCode	String(6)	No	Authorization code provided by the issuing bank
loyaltyAmount	String(20)	No	Amount in loyalty points for combined transactions (RON + LOY)
loyaltyLink	String(32)	No	orderNumber of the transaction in LOY for combined transactions (RON + LOY)
status	String(20)	No	Payment status
date	String(20)	No	The date the transaction was recorded
paymentDate	String(100)	No	Date of payment
merchantFullName	String(512)	No	Merchant name
bankName	String(200)	No	The name of the issuing bank, if it could be identified
maskedPan	String(12)	No	Masked card number
panCountryCode	String(2)	No	Card issuing country

7 Payment in installments and/or in loyalty points

7.1 Payment page with installments

The option to pay in instalments is displayed only if the merchant has an installment payment acceptance agreement via the StarCard program, signed with Banca Transilvania. Furthermore, the paying customer must use a **credit card issued by Banca Transilvania** during the payment process.

On the payment page, in the “Pay in instalments” field the following options will be displayed:

- “Full payment” – for full payment
- The number of installments, according to the aggreement signed between the merchant and Banca Transilvania

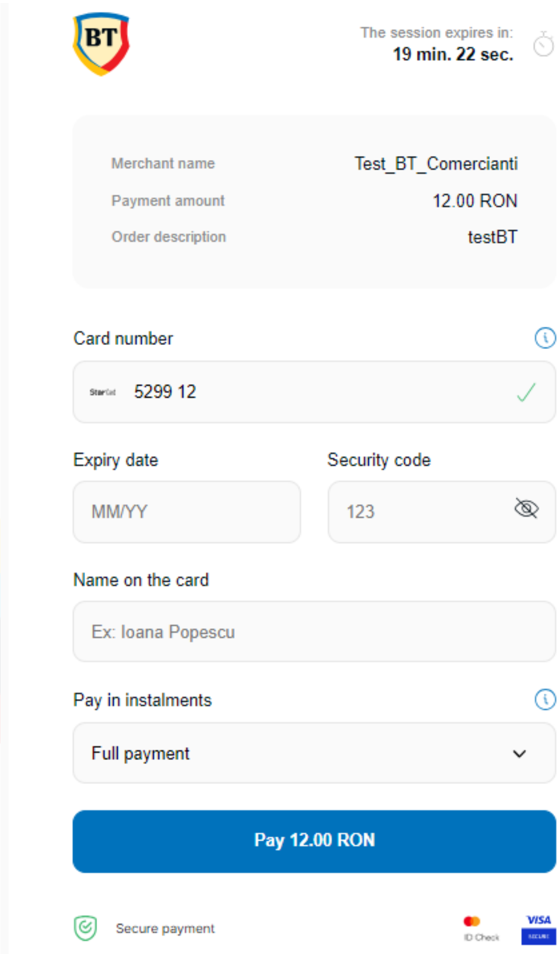

Figure 6. Payment page with installments

7.2 Payment page with installments and loyalty points

The option to pay in instalments and loyalty points is displayed only if the merchant has an installment payment acceptance agreement and usage of loyalty points via the StarCard program, signed with Banca Transilvania. Furthermore, the paying customer must use a **credit card issued by Banca Transilvania** during the payment process.

On the payment page, in the "Pay in instalments" field the following options will be displayed:

- "Full payment" - for full payment
- The number of installments, according to the aggreement signed between the merchant and Banca Transilvania

On the payment page, in the field "Pay with Star points", the client will set the desired number of loyalty points to be used in the payment. The equivalent monetary value of these points will be deducted from the total amount due.

Attention! Star points cannot be used for transactions in other currencies (EUR, USD), but only in currency RON. The number of points cannot exceed the amount of the transaction nor the value of 1500 starBT points (LOY).

Attention! Payment in starBT points is not processed in installments. For a transaction made in RON + LOY, only the amount of the transaction in RON will be processed in installments.

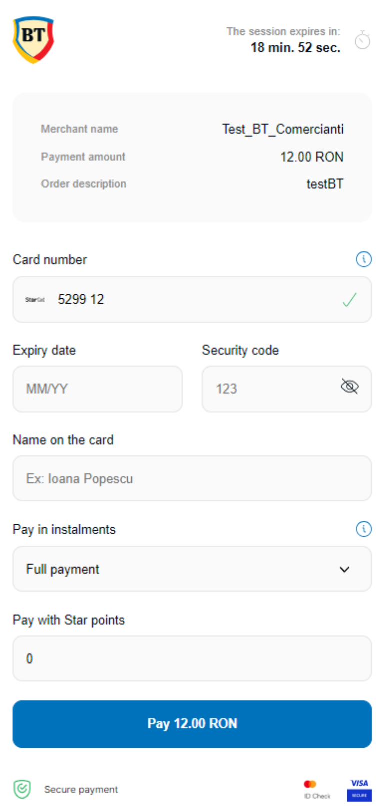

Figure 7. Payment page with installments and loyalty points

7.2.1 Approved transaction

The initial transaction will be divided into two transactions, as follows: the value of the transaction in RON, and the second transaction of the value of the number of starBT points (LOY). The number of points entered by the customer will be processed in RON.

These two transactions will have differents UUID (mdOrder) in the iPay system. To identify the transaction made in starBT points, use the API getOrderStatusExtended.do and the mdOrder of the transaction in RON. Additional information is:

a) The *merchantOrderParams* block will additionally display:
- the name parameter with the *loyaltyAmount* value and the value parameter with the value of starBT points in cash

b) The *attributes* block will display in addition:
- the name parameter with the *loyalties* value and the value parameter with the *mdOrder* and *orderNumber* of the transaction in LOY
- the name parameter with the *installment* value and the value parameter with the number of installments selected by the client. They are missing if the payment was not made in installments.

7.2.2 Declined transaction

If the client does not have the selected points on his account, the transaction in RON is automatically reversed and will have REVERSED status, and the transaction in LOY will have DECLINED status.

If the transaction is declined for various reasons (Card blocked, insufficient funds, limit exceeded, wrong CVV, expired card, cardholder's issuing bank declines the transaction, etc), only the transaction in RON with DECLINED status will be registered.

7.2.3 Deposit

For payments made with 2-phase and APPROVED status, the collection is made by calling the deposit.do API for both the LOY transaction and the RON transaction. This operation can only be performed once.

7.2.4 Reverse and refund

The **reversal** of the transactions made through 2-phase and with APPROVED status, is done by calling the reverse.do API, both for the LOY transaction and for the RON transaction. This operation can only be performed once.

The **refund** of the transactions is done by calling the refund.do API and for transactions with DEPOSITED status.

Attention! For a partial refund, the LOY part is refunded for the first time.

We have a transaction of 10 RON, divided into 6 RON + 4 LOY and we want to refund:

• the amount of 2 LEI => we make refund on 2 LOY

• the amount of 7 LEI => we make refund on 4 LOY + 3RON

8 Appendix: Other error codes

Other error codes for the actionCode parameter and their description.

Action code	error_id	error_message	Description
-20010	-20010	BLOCKED_BY_LIMIT	Transaction is rejected since the amount exceeds the limits specified by the Issuing bank
-9000	-9000	Started	State of the transaction start
-3003	-3003	Unknown	Unknown
-2102	-2102	Blocking by a passenger name.	Reject by the passenger name
-2101	-2101	Blocking by e-mail	Reject by email
-2020	-2020	Incorrect ECI received	Invalid ECI. This code means that ECI received in PaRes is not valid for the IPN. The rule applies only to Mastercard (the available values - 01,02) and Visa (the available values - 05,06)
-2019	-2019	Decline by iReq in PARes	PARes from the issuing bank contains iReq, which caused the payment rejection
-2018	-2018	Declined. DS connection timeout	There is no access to the Directory server Visa or MaterCard; or a connection error occurred after a card involvement request (VeReq). This is an error of interaction between the payment gate and IPN servers due to technical problems on the side of IPN servers.
-2017	-2017	Declined. The PARes status is not "Y"	Rejected. PARes status is not "Y"
-2016	-2016	Declined. VeRes status is unknown	Issuing bank could not determine if the card is 3-D Secure.
-2015	-2015	Decline by iReq in VERes	VERes from DS contains iReq, which caused the payment rejection.
-2013	-2013	All attempts to perform a payment are used.	All payment attempts were used.
-2012	-2012	Operation not supported	This operation is not supported.
-2011	-2011	Declined. PaRes status is unknown	Issuing bank was not able to perform the 3-D Secure card authorization.
-2010	-2010	XID mismatch	XID mismatch.
-2008	-2008	Incorrect wallet	Wrong wallet.
-2007	2007	Decline. Payment time limit	The period allotted to enter the card details has expired (by default the timeout is 20 minutes; the session duration may be specified while registering an order; if the merchant has "Alternative session timeout" permission, then the timeout duration is specified in merchant settings).

Action code	error_id	error_message	Description
-2006	2006	Decline. 3DSec decline	Means that the issuing bank rejected authentication (3DS authorization has not been performed).
-2005	2005	Decline. 3DSec sign error	Issuing bank sign could not be checked, i.e. PARes was readable but the sign was wrong.
-2003	-2003	Blocking by the port	Blocking by the port.
-2002	2002	Decline. Payment over limit	Transaction was rejected because the payment amount exceeded the established limits. Note: it could be the limit of the day withdrawal established by the acquiring Bank, or the limit of transactions by one card established by the merchant, or the limit for one transaction established by a merchant.
-2001	2001	Decline. IP blacklisted	Transaction is rejected since Client's IP-address is in the black list.
-2000	2000	Decline. PAN blacklisted	Transaction is rejected since the card number is in the black list.
-102	-102	A payment is cancelled by the payment agent	The payment was cancelled by the payment agent.
-100	-100	no_payments_yet	There were not payment attempts.
-1	-1	sv_unavailable	The timer of waiting for a processing response has expired.
0	0	Approved.	Payment has been performed successfully.
1	1	Declined. Honor with id	Proof of identity is necessary for successful completion of the transaction. In case of an internet transaction (our case) it is impossible, so the transaction is considered as declined.
5	5	Decline. Unable to process	Rejection of the network to process a transaction.
100	100	Decline. Card declined	Card limits (the Issuing bank forbade internet transactions by the card).
101	101	Decline. Expired card	Card is expired.
103	103	Decline. Call issuer	There is no connection to the Issuing bank. The sales outlet needs to contact the Issuing bank.
104	104	Decline. Card declined	This is an attempt to perform a transaction by the account that has restrictions for use.
106	106	The maximum number of attempts to enter PIN is exceeded. The card might be temporary blocked.	The maximum number of attempts to enter PIN is exceeded. It is possible that the card is blocked temporary.
107	107	Decline. Call issuer	Please, contact the Issuing bank.
109	109	Decline. Invalid merchant	Merchant/terminal identifier is incorrect or ACC is blocked at the processing level.

Action code	error_id	error_message	Description
110	110	Decline. Invalid amount	Transaction amount is incorrect.
111	111	Decline. No card record	Card number is incorrect.
116	116	Decline. Not enough money	Transaction amount exceeds the available balance of the selected account.
117	117	INCORRECT PIN	Incorrect PIN (not for internet transactions).
119	119	Decline. SECURITY_VIOLATION from processing system	Illegal transaction.
120	120	Decline. Not allowed	Refusal to perform the operation - the transaction is not allowed by the Issuing bank. The response code of the IPN is 57. Reasons for rejection should be specified by the issuing bank.
121	121	Decline. Excds wdrwl limt	This is an attempt to perform a transaction of the amount exceeding the day limit established by the issuing bank.
123	123	Decline. Excds wdrwl ltmt	The client has performed the maximum number of transactions during the limit cycle and tries to perform another one.
125	125	Decline. Card declined	Card number is incorrect. This error may have several meanings: An attempt to perform a refund of the amount exceeding the hold amount; An attempt to refund a zero amount; for AmEx – the expiry date is specified incorrectly.
208	208	Decline. Card is lost	Card is lost.
209	209	Decline. Card limitations exceeded	Card limitations exceeded.
400	400	The reversal is processed.	Reversal is processed.
902	902	Decline. Invalid trans	Card limitations (the cardholder tries to perform a transaction that is forbidden for him).
903	903	Decline. Re-enter trans.	Attempt to perform a transaction of amount exceeding Issuing bank limit.
904	904	Decline. Format error	The message format is incorrect in terms of the issuing bank.
907	907	Decline. The host is not available.	There is no connection to the Issuing bank. Authorization in the stand-in mode is not allowed for this card number (this mode means that the Issuing bank is unable to connect to the IPN, and therefore the transaction can be either offline with further unloading to back office, or it can be declined).

Action code	error_id	error_message	Description
909	909	Decline. Call issuer	Operation is impossible (a general error of the system functioning. May be detected by IPN or the Issuing bank).
910	910	Decline. The host is not available.	Issuing bank is not available.
913	913	Decline. Invalid trans	The message format is incorrect in terms of IPN.
914	914	Decline. Orig trans not found	Transaction is not found (when sending a completion, reversal or refund request).
999	999	Declined by fraud	The beginning of the transaction authorization is missed. Declined by fraud.
1001	1001	Decline. Data input timeout	Empty (is specified at the moment of the transaction authorization, when card details are not entered yet).
1004	1004	Authorization phase 1	Authorization phase 1.
1005	1005	Authorization phase 2	Authorization phase 2.
2001	2001	Decline. Fraud	Fraud (in terms of IPN).
2002	2002	Incorrect operation	Incorrect operation.
2003	2003	Decline. SSL restricted	SSL (not 3-D Secure/SecureCode) transactions are forbidden for the Merchant.
2004	2004	SSL without CVC forbidden	Payment through SSL without CVC2 is forbidden.
2005	2005	3DS rule failed	Payment does not meet the terms of the rule of 3-D Secure validation.
2006	2006	One-phase payments are forbidden	One-phase payments are forbidden.
2007	2007	The order is paid	The order is paid.
2008	2008	The transaction is not completed	The transaction is not completed.
2009	2009	Refund amount exceeds the payment amount	Refund amount exceeds the deposited amount.
2014	2014	3-D Secure rule execution error	Error of a 3-D Secure rule execution.
2015	2015	Terminal select rule error	Terminal select rule error (a rule is incorrect).
9001	9001	Internal error	Internal error.
71015	1015	Decline. Input error	Entered card details are incorrect.
151017	1017	Decline. 3DSec comm error	3-D Secure - communication error.
151018	018	Decline. Processing timeout	Processing timeout. Sending is failed.
151019	1019	Decline. Processing timeout	Processing timeout. Sending is successful; a response from the bank was not received.
341014	1014	Decline. General Error	General error.

9 Contact

If you have questions or encounter problems during implementation, please contact us at the email address aplicatiiecommerce@btrl.ro .