bankart

UNLIMITED

THE USE OF BANKART'S PAYMENT GATEWAY – INSTRUCTIONS FOR MERCHANTS

March 2025

UNLIMITED

Table of contents

1. **GLOSSARY**

Authentication (3DS): The process of verifying a user's identity to ensure that the user has the right to carry out a transaction.

Authorization: The process of verifying card details, limits and whether the customer has sufficient funds in their bank account to make the purchase they wish to make, etc. This process generally occurs after authentication has been successfully completed, but in certain exceptions even without successful authentication.

Payment Gateway: A system that enables or simplifies the acceptance of card payments or instant payments (Flik, etc.) to merchants. It is a system that connects websites (stores) with issuers and acquirers of payment methods. Payment Gateway also provides a connection to the customer authentication systems. In addition, Payment Gateway has a web interface to facilitate the review of transactions made.

Access Control Server (ACS): A system used by the card-issuing banks for authentication of cardholders using several different methods.

2. INTRODUCTION

This document is intended for merchants who use Bankart's Payment Gateway to accept card payments. The document provides the information necessary for the integration of Payment Gateway, in addition to general information on card transactions and testing instructions.

2.1. Online transaction flow

Below is a schematic diagram of an indicative online transaction flow, with a description of each step of the online transaction underneath for clarity.

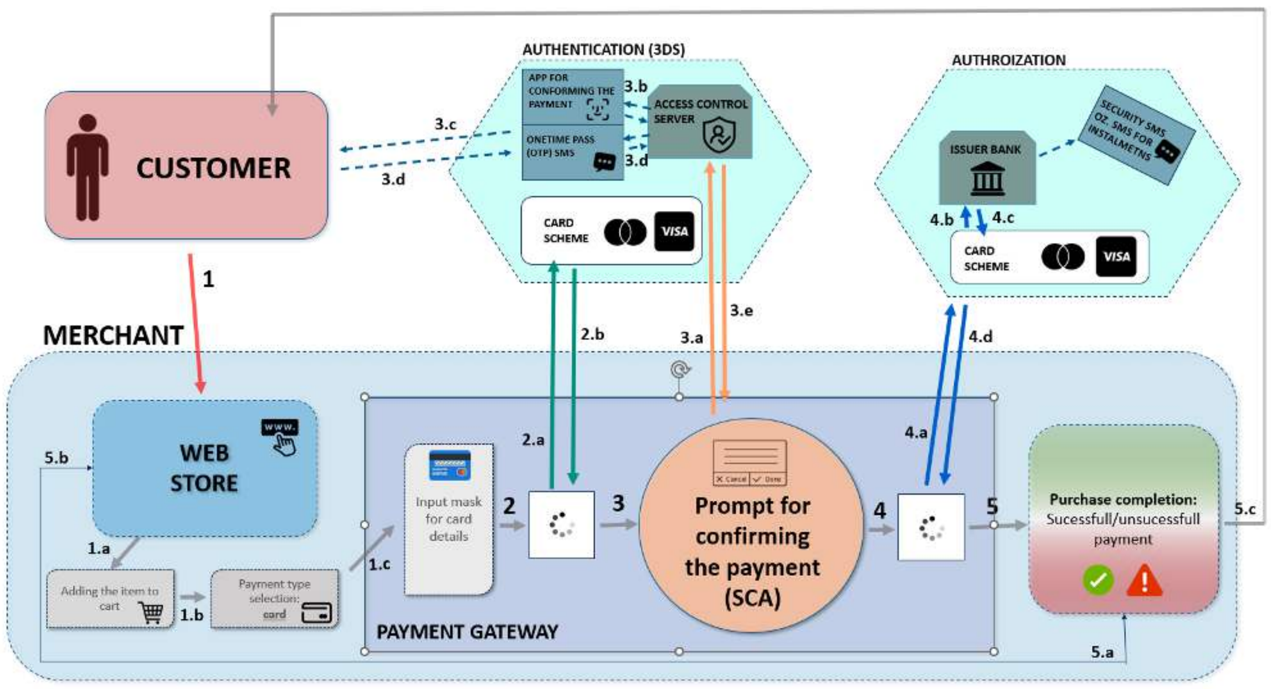


Figure 1: Online transaction flow

Online transaction flow steps:

1. A **customer** browses an online shop looking for products:
   a. adds products to the basket;
   b. selects a payment method, which in this case is a payment card, and confirms the purchase;
   c. the merchant makes an API call to **Bankart's Payment Gateway** and an input mask appears to enter the customer's card details.
2. After entering the card details, their authenticity is verified (*Card Enrollment*):
   a. Payment Gateway connects to the card scheme to obtain data about the card, the version of the 3DS protocol and data on whether the card is even included in the 3DS system;
   b. the card scheme forwards the data back to Payment Gateway.
3. Payment Gateway connects to the *payment authentication* system:
   a. if the transaction is a transaction for which strong customer authentication is not required, the transaction is executed without strong customer authentication;
   b. if strong authentication is required, the ACS sends an authentication request to the customer's payment confirmation app (usually in the form of a push notification) or

a unique password (OTP) is sent to the phone number;

c. the customer performs authentication;
d. the authentication system transmits the authentication result information to the ACS system;
e. The ACS forwards the authentication result information to Payment Gateway.

4. In the next step, the *authorization* process starts:
   a. Payment Gateway sends an authorization request to the card scheme;
   b. the latter forwards the request to the customer's bank, where the purchase is authorized;
   c. the customer's bank returns the response to the card scheme, and at the same time the process is initiated for sending a security text message (if the service is activated) and an instalment text message (if it is an instalment purchase) to the customer;
   d. the card scheme returns a response to Payment Gateway.
5. Payment Gateway calls the URL to display the transaction success screen:
   a. Payment Gateway returns the transaction result to the merchant;
   b. the merchant acknowledges the receipt of the result;
   c. the customer is presented with the transaction success/failure screen.

3. TECHNICAL INTEGRATION OF THE CARD PAYMENT METHOD (INSTRUCTIONS FOR DEVELOPERS)

3.1. Main steps in the implementation of support

1. Signing a contract between the merchant and the bank;
2. Transmission of data from the bank to Bankart, where the merchant is entered into the online business support system;
3. Forwarding of the data for the purposes of implementing support to the merchant's contact mobile phone number and email address (passwords for access to the Payment Gateway portal and API/WEB connection of the user);
4. Merchant's developers perform integration with Payment Gateway;
5. Performance of tests covering the agreed scope of business operations (authorization, capture, void, instalments, correct display of notifications when a transaction is authorized and declined, etc.);
6. The bank registers the merchant via MasterCard Connect in the EMV 3D Secure system. No registration is required in the case of Visa, as the data is already automatically sent to the Bankart systems upon login;
7. The merchant notifies Bankart of the successful completion of the tests, Bankart further verifies the results in terms of
   process centre support;
8. Forwarding the data for accessing the production environment to the merchant.

3.2. Technical documentation and requirements

General documentation on Payment Gateway is available at: https://gateway.bankart.si/documentation/gateway
API documentation on connecting and integration is available at: https://gateway.bankart.si/documentation/apiv3?php#transaction-api-v3-0 (JSON)

All API calls must contain a signature that is generated based on the content sent and the secret shared. Callbacks are also signed so you can verify the authenticity of the received content. API calls can be modified on the Payment Gateway side in some of the optional fields. Merchant-side support must be done in such a way that the change does not affect the functioning. It is necessary to emphasize that when initiating a transaction with an API call in JSON, the merchant must not forget to send the callbackURL field, because the result of the transaction will be sent to this URL address when it is completed.

Sample code in programming languages PHP, Java, C# (.NET Framework) and plugins for the latest versions of WooCommerce and Wordpress (v4 and v5), Prestashop (v1.7 – previous versions are not supported), Magento (v2) and Opencart (v3 – v2 is not supported) platforms is available at: link.

3.3. Merchant-side transaction processing support

SSL communication is used for communication between the merchant's website and Payment Gateway. For this purpose, the merchant must obtain a suitable, generally supported server certificate on their side.

IMPORTANT: the merchant or their IT support must ensure that the server certificate is replaced on time before expiry, as proper support is not possible without a valid certificate. Monitoring production after the change is made on the side of the merchant is required. If the merchant does not obtain the same processing results at the time of monitoring as before the change was made, the merchant should revert their system to the previous state and inform Bankart (customer.support@bankart.si).

3.4. Obtaining card details from the customer

Payment Gateway supports different ways of retrieving card details from the customer. The merchant can use a pre-prepared payment page with an input mask or integrate it into their own website. It is also possible to store payment card data, whereby sensitive card data is stored on the Payment Gateway site and the merchant handles only the token of the individual card. A more detailed description is given below.
For the authentication and authorization process of the transaction, the customer's details listed below are important and are obtained by the merchant from the user account or upon entering contact details before entering the card details. It is recommended that the merchant verifies whether the customer has entered the details in the correct format, as this makes the transaction more likely to be successful.

Name of field	Description
billAddrCity	Place of residence. Format: variable length, 50 characters maximum
billAddrCountry	Country of residence. Format: 3 characters maximum, JSON Data Type: String, necessary to fill in with a numerical 3-digit code according to the ISO 3166-1 standard (exceptions are provided in Table A.5 in the EMV 3DS specification)
billAddrLine1	Address. Format: variable length, 50 characters maximum, Data Type: String
billAddrPostCode	Postal code of residence. Format: variable length, 16 characters maximum, Data Type: String
email	Email address of the payer. Format: variable length, 254 characters maximum, necessary to fill in with data meeting the requirements provided in the table in Section 3.4 of document IETF RFC 5322.
cardholderName	Name of the cardholder. Format: variable length, 2–45 characters (verified by Bankart)

3.4.1. Designing a payment page with an input mask

We can add the merchant's logo to the Hosted Payment Page (HPP), which displays an input mask for the card details and is displayed in case of a redirect from the merchant's page. This should be submitted by the merchant in *.png* format to customer.support@bankart.si prior to inclusion in the production. The maximum image size is *500px x 130px*. Please contact your bank for logos of the payment schemes involved.
Currently, the HPP site can be displayed in 19 different languages. These languages are Slovenian, English, German, Italian, Bulgarian, Bosnian, Czech, Slovak, Spanish, Croatian, Hungarian, Montenegrin, Macedonian, Polish, Romanian, Russian, Albanian, Serbian and Turkish. For additional language options, please contact your bank.

3.4.2. Custom payment page

For merchants who want a customized payment page, we suggest the **payment.js integration**, which is described in the API documentation freely available to developers on the website: https://gateway.bankart.si/documentation/gateway#paymentjs-javascript-integration

If the merchant is transitioning from the HPP page to payment.js integartion or decides to implement it at the beginning of the integration, additional tests of the API calls enabling this implementation are required. To arrange this process, the merchant should contact Bankart's customer support.

3.5. Types of transactions supported on Payment Gateway

The merchant agrees with the bank and specifies in the contract which types of transactions they wish to use. They then implement and test only those transactions that the bank has approved in accordance with the prior agreement.

3.5.1. Basic types of transactions supported:

- **DEBIT (normal online purchase):**
  This type of one-step transaction is typically used by merchants selling digital goods, tickets and other items that are available to customers immediately after payment is completed. The merchant cannot void these (final) authorizations of amounts that are marked to be included in the end-of-day clearing and settlement processing.
- **REFUND (money back for online purchases):**
  If a merchant wants to make a refund due to customer dissatisfaction or other issues, they must initiate this type of transaction. Refund transactions must always have a reference to a financial transaction (debit, capture) and must not exceed the amount of the original transaction.

  DEBIT + REFUND scheme:

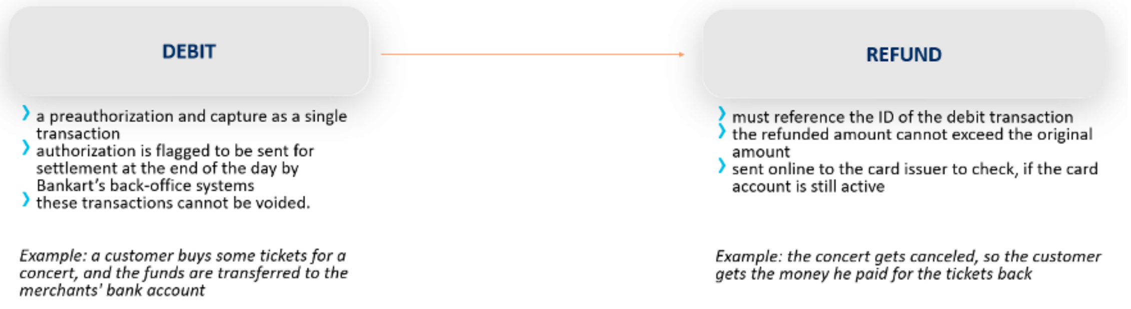
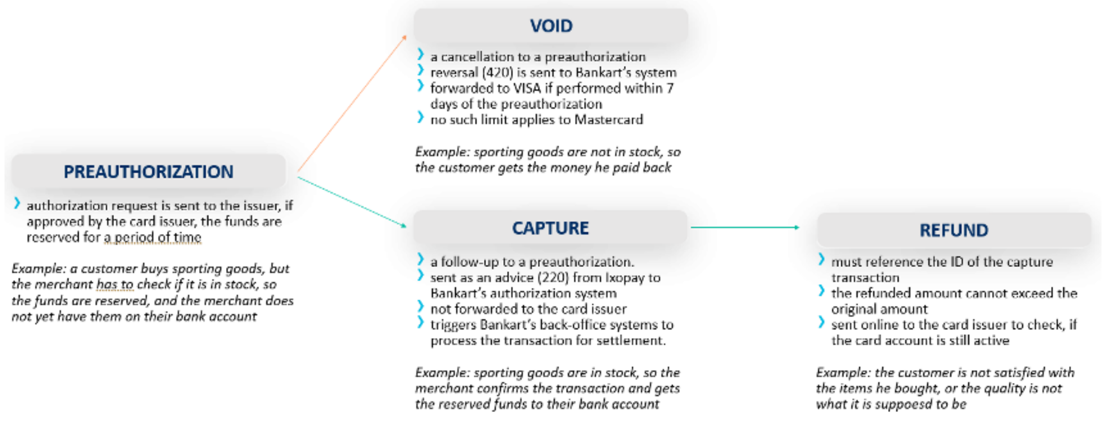

DEBIT
a preauthorization and capture as a single transaction
authorization is flagged to be sent for settlement at the end of the day by Bankart's back-office systems
these transactions cannot be voided.
Example: a customer buys some tickets for a concert, and the funds are transferred to the merchants' bank account
REFUND
must reference the ID of the debit transaction
the refunded amount cannot exceed the original amount
sent online to the card issuer to check, if the card account is still active
Example: the concert gets canceled, so the customer gets the money he paid for the tickets back

Figure 2: Display of the DEBIT+REFUND flow

- **PREAUTHORIZE + CAPTURE and VOID + REFUND:**
  A merchant selling physical goods may first want to check their stock before confirming payment. If the merchant finds that they cannot fulfil the order, they can cancel the authorization and release the reserved funds in the cardholder's account (perform a VOID). This is why we call it a two-step transaction.
  The transaction is thus split into two parts, the (pre)authorization and the subsequent CAPTURE command, which indicates that the transaction should be included in the end-of-day clearing and settlement processing. The capture should be done within one week (depending on the card scheme) to avoid a higher risk of chargebacks. Once a transaction has been captured, it cannot be voided, but a REFUND (money back) can be made. If a partial capture is made, a VOID must be done for the remaining amount to close the original pre-authorization and ensure the customer receives the remaining amount they paid.

PREAUTHORIZE + CAPTURE/VOID + REFUND scheme:



PREAUTHORIZATION

› authorization request is sent to the issuer, if approved by the card issuer, the funds are reserved for a period of time

Example: a customer buys sporting goods, but the merchant has to check if it is in stock, so the funds are reserved, and the merchant does not yet have them on their bank account

VOID

› a cancellation to a preauthorization
› reversal (420) is sent to Bankart's system
› forwarded to VISA if performed within 7 days of the preauthorization
› no such limit applies to Mastercard

Example: sporting goods are not in stock, so the customer gets the money he paid back

CAPTURE

› a follow-up to a preauthorization.
› sent as an advice (220) from Ixopay to Bankart's authorization system
› not forwarded to the card issuer
› triggers Bankart's back-office systems to process the transaction for settlement.

Example: sporting goods are in stock, so the merchant confirms the transaction and gets the reserved funds to their bank account

REFUND

› must reference the ID of the capture transaction
› the refunded amount cannot exceed the original amount
› sent online to the card issuer to check, if the card account is still active

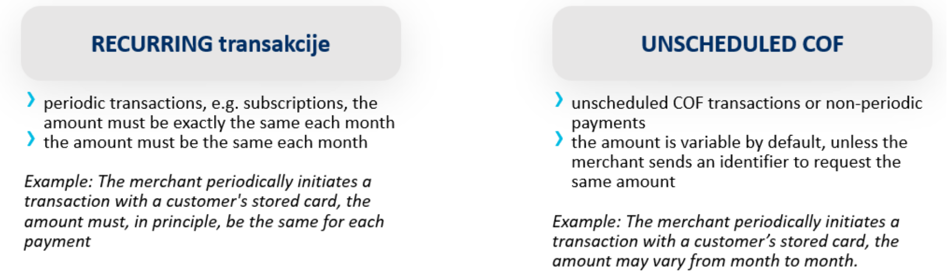
Example: the customer is not satisfied with the items he bought, or the quality is not what it is suppoesd to be

Figure 3: Display of the PREAUTH+VOID/CAPTURE flow

3.5.2. Types of supported transactions with card-storing capability

These transaction types allow merchants to reuse card details for future transactions (COF or Card on File) without requiring the cardholder to re-enter the card number and expiry date. Both transactions are initiated by the cardholder, with the cardholder performing strong authentication upon data entry. The merchant must first sign a contract with the bank, by means of which the bank enables them to carry out transactions with the stored card in the first place. Only once the legal formalities are in place can the merchant mark, test and offer transactions with a stored card.

- **CARDHOLDER INITIATED TRANSACTIONS (CIT):**
  The merchant offers the cardholder the option of payment with the stored card. The merchant receives a masked PAN in the original transaction when the card was stored and can use it to display and offer the stored payment credentials to the cardholder.
  These transactions are the same as normal transactions, except that in subsequent transactions the cardholder does not manually enter the card number but retrieves it from the secure storage system. The cardholder is prompted for authentication via EMV 3DS for the initial transaction, as for any other online purchase.
- **MERCHANT INITIATED TRANSACTIONS (MIT):**
  In the context of e-commerce, RECURRING transactions are meant as, e.g. periodic charges for magazine subscriptions. In recurring transactions, the amount is the same each time.
  Another type of MIT transaction is the UNSCHEDULED Card on File (COF) transaction, which means non-periodic payments that are made with a variable amount by default, unless the merchant sends an additional indicator in the message to request the same amount.



RECURRING transakcije

› periodic transactions, e.g. subscriptions, the amount must be exactly the same each month
› the amount must be the same each month

Example: The merchant periodically initiates a transaction with a customer's stored card, the amount must, in principle, be the same for each payment

UNSCHEDULED COF

› unscheduled COF transactions or non-periodic payments
› the amount is variable by default, unless the merchant sends an identifier to request the same amount

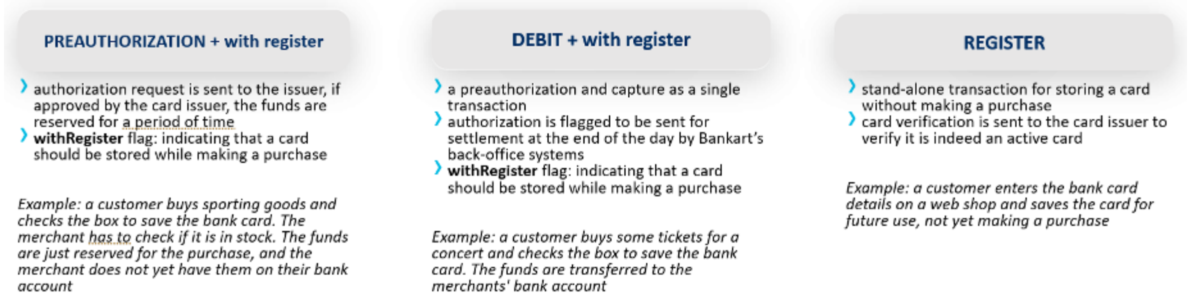
Example: The merchant periodically initiates a transaction with a customer's stored card, the amount may vary from month to month.

Figure 4: Scheme of division of recurring payments

3.5.3. Card storing

A card can be stored in a number of ways:

- **<u>REGISTER (storing a card without a purchase):</u>**
  The customer enters the card details at the online store, which offers the option to store the card. First, the customer is authenticated and then a request for confirmation of the card activity is sent to the card issuer. After confirmation, a reference of this transaction is stored for further use of the card.
- **<u>DEBIT with the "withRegister" flag (normal online purchase with an additional indicator):</u>**
  A classic online purchase, where the customer chooses to save the card for future purchases at the time of payment. The customer has to authenticate themselves and the card details are stored in Bankart's Payment Gateway, as is the reference to this first transaction.
- **<u>PREAUTHORIZE with the "withRegister" flag (preauthorization with an additional indicator):</u>** A merchant selling physical goods may first want to check stock before confirming payment. When entering the card details, the customer also selects the option to save the selected card for future purchases. The customer has to authenticate themselves and the card details and the reference to the first transaction are stored in Bankart's Payment Gateway.
- **<u>DEREGISTER (removal of a stored card)</u>**
  This transaction can be executed by either the merchant or the customer. This type of transaction removes a card already stored with a merchant. The customer can remove the card if they no longer wish to use the card at the merchant, and the merchant can remove the card if the customer stores a new card without deleting the old card that they no longer use. The merchant is responsible for deleting all card data that is no longer needed (also in the event of the business ceasing its activities). In order to perform this transaction, a reference to the first transaction in the series must be sent within the API call (the UUID reference field mentioned in the next subsection).



PREAUTHORIZATION + with register

› authorization request is sent to the issuer, if approved by the card issuer, the funds are reserved for a period of time
› **withRegister** flag: indicating that a card should be stored while making a purchase

Example: a customer buys sporting goods and checks the box to save the bank card. The merchant has to check if it is in stock. The funds are just reserved for the purchase, and the merchant does not yet have them on their bank account

DEBIT + with register

› a preauthorization and capture as a single transaction
› authorization is flagged to be sent for settlement at the end of the day by Bankart's back-office systems
› **withRegister** flag: indicating that a card should be stored while making a purchase

Example: a customer buys some tickets for a concert and checks the box to save the bank card. The funds are transferred to the merchants' bank account

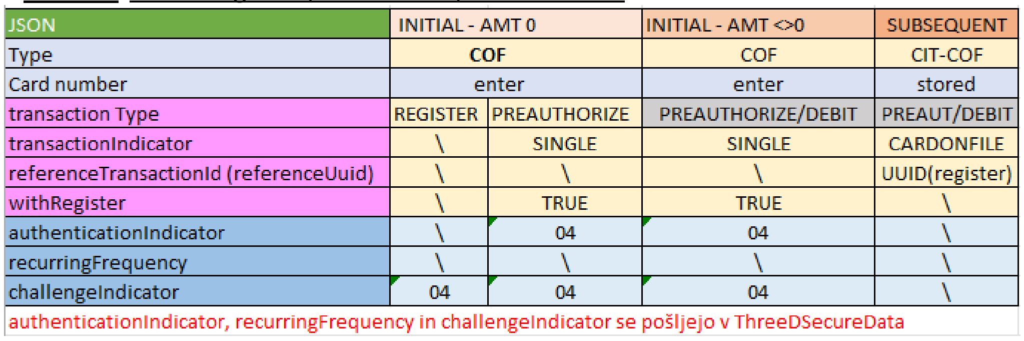
REGISTER

› stand-alone transaction for storing a card without making a purchase
› card verification is sent to the card issuer to verify it is indeed an active card

Example: a customer enters the bank card details on a web shop and saves the card for future use, not yet making a purchase

The following shows how the different types of transactions that have the possibility of storing a card (CoF, MiT and recurring) should be indicated. Tables indicating the main indicators are enclosed, as well as examples of JSON calls.

<u>3.5.3.1. Indicating CoF (Card on File) transactions:</u>



JSON	INITIAL - AMT 0		INITIAL - AMT <>0	SUBSEQUENT
Type	COF		COF	CIT-COF
Card number	enter		enter	stored
transaction Type	REGISTER	PREAUTHORIZE	PREAUTHORIZE/DEBIT	PREAUT/DEBIT
transactionIndicator	\	SINGLE	SINGLE	CARDONFILE
referenceTransactionId (referenceUuid)	\	\	\	UUID(register)
withRegister	\	TRUE	TRUE	\
authenticationIndicator	\	04	04	\
recurringFrequency	\	\	\	\
challengeIndicator	04	04	04	\
authenticationIndicator, recurringFrequency in challengeIndicator se pošljejo v ThreeDSecureData				

Figure 5: Table with indicators for CoF transactions

Within CoF transactions, there are 3 types of possible implementations of these transactions:

- **<u>Initial</u>**<u>, where the amount equals 0:</u>
  It is a customer-initiated transaction (CIT) where card details have to be entered.
  - In the "*transaction Type*" field, the transaction should be marked as REGISTER, which will only save the card. Alternatively, PREAUTHORIZE can be selected, where the indicators then differ from the REGISTER transaction.
  - In the case of a REGISTER transaction, the "withRegister" field is empty. In the case of a PREAUTHORIZE transaction, the value in the "withRegister" field is TRUE.
  - The field "*authenticationIndicator*" is set to 04 in the case of a PREAUTHORIZE transaction. In the case of a REGISTER transaction, the field is left empty. The value of "04" means that the card is only added, without setting up a recurring/MIT series.
  - The field "*challengeIndicator"* is populated with the value "04", which means that strong authentication (SCA) is required by the card schemes for each initial transaction.
  - Other fields marked above do not need to be filled in.

- **<u>Initial</u>**<u>, where the amount is different from 0:</u>
  It is a customer-initiated transaction (CIT) where card details have to be entered.
  - The "*transaction Type*" field is marked as DEBIT or PREAUTHORIZE, depending on which transaction type the merchant supports or wishes to perform.
  - In the "*transaction Indicator*" field, the transaction should be marked as SINGLE. This is to note that it is a stand-alone transaction or purchase (no series is created), but also to store the card details. This is marked TRUE in the "*withRegister*" field.
  - The "*authenticationIndicator*" field is set to 04 for this type of transaction implementation, as it is linked to the "*transaction Type*" field.
  - The field "*challengeIndicator"* is populated with the value "04", which means that strong authentication (SCA) is required by the card schemes for each initial transaction.
  - Other fields marked above do not need to be filled in.

- **<u>Subsequent</u>** <u>transactions with the stored card:</u>
  It is a transaction initiated by the customer (CIT) using a previously stored card.
  - In the "*transaction Type*" field, the transaction needs to be marked as DEBIT or PREAUTHORIZE, depending on which transaction type the merchant supports or wishes to perform.
  - In the "*transaction Indicator*" field, the transaction should be marked as CARDONFILE, thus indicating that it is a subsequent transaction, but linked to the first one.
    - Therefore, the field "*referenceTransactionId*" should be filled with the UUID (unique indicator) of the Register or the initial transaction, because the system thus knows that strong authentication has already been performed on the first transaction and does not need to be performed here.

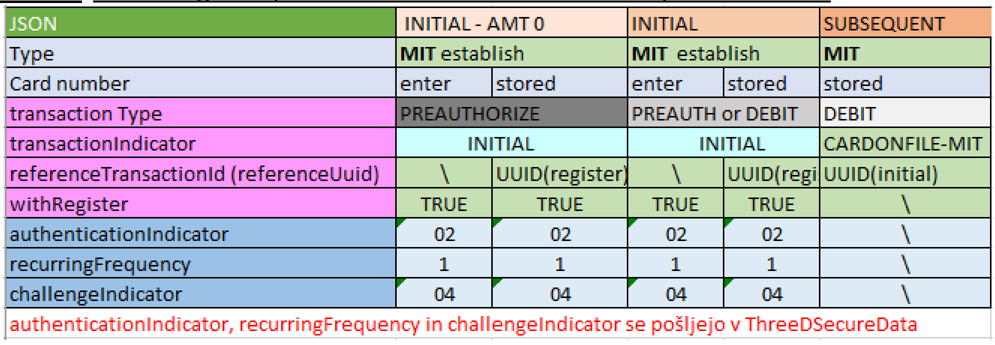
Example of a code for INITIAL (first) transactions, in JSON format:

```
{
   "merchantTransactionId": "D-2022-04-07-624eb695e63d3",
   "extraData": {
      "userField1": "00",
   },
   "threeDSecureData":{
       "authenticationIndicator":"04"
   }
   "merchantMetaData": "Transaction:Debit;Description:test",
   "amount": "1.99",
   "currency": "EUR",
   "successUrl": " https://mydomain.com/PHPPaymentGatewayJson/examples/PaymentOK.php",
   "cancelUrl": " https://mydomain.com/PHPPaymentGatewayJson/examples/PaymentCancel.php",
   "errorUrl": " https://mydomain.com/PHPPaymentGatewayJson/examples/PaymentNOK.php",
   "callbackUrl": " https://mydomain.com/PHPPaymentGatewayJson/examples/Callback.php",
   "description": "One pair of shoes",
   "withRegister": true,
   "transactionIndicator": "SINGLE",
   "customer": {
      "firstName": "Janez",
      "lastName": "Novak",
      "billingAddress1": "Street 1",
      "billingCity": "City",
      "billingPostcode": "1000",
      "billingCountry": "SI",
      "email": "test@email.com",
      "ipAddress": "91.208.168.48"
   },
   "language": "sl"
}
```

Example of a code for SUBSEQUENT (further) transactions, in JSON format:

```
{
   "merchantTransactionId": "D-2022-04-07-624ebfa95f6d9",
   "extraData": {
      "userField1": "00"
   },
   "merchantMetaData": "Transaction:Debit;Description:test",
   "referenceUuid": "a4189ed3a0ea2546826a",
   "amount": "1.99",
   "currency": "EUR",
   "successUrl": https://mydomain.com/PHPPaymentGatewayJson/examples/PaymentOK.php",
   "cancelUrl": " https://mydomain.com/PHPPaymentGatewayJson/examples/PaymentCancel.php",
   "errorUrl": " https://mydomain.com/PHPPaymentGatewayJson/examples/PaymentNOK.php",
   "callbackUrl": "https://mydomain.com/PHPPaymentGatewayJson/examples/Callback.php",
   "description": "One pair of shoes",
   "withRegister": false,
   "transactionIndicator": "CARDONFILE",
   "customer": {
      "firstName": "Janez",
      "lastName": "Novak",
      "billingAddress1": "Street 1",
      "billingCity": "City",
      "billingPostcode": "1000",
      "billingCountry": "SI",
      "email": "test@email.com",
      "ipAddress": "91.208.168.48"
   },
   "language": "sl"
}
```

3.5.3.2. Indicating MIT (Merchant Initiated Transactions) transactions:



JSON	INITIAL - AMT 0		INITIAL		SUBSEQUENT
Type	**MIT** establish		**MIT** establish		**MIT**
Card number	enter	stored	enter	stored	stored
transaction Type	PREAUTHORIZE		PREAUTH or DEBIT		DEBIT
transactionIndicator	INITIAL		INITIAL		CARDONFILE-MIT
referenceTransactionId (referenceUuid)	\	UUID(register)	\	UUID(regi	UUID(initial)
withRegister	TRUE	TRUE	TRUE	TRUE	\
authenticationIndicator	02	02	02	02	\
recurringFrequency	1	1	1	1	\
challengeIndicator	04	04	04	04	\
authenticationIndicator, recurringFrequency in challengeIndicator se pošljejo v ThreeDSecureData					

Figure 6: Table with indicators for MIT transactions

Within MIT transactions, there are 3 types of possible implementations of these transactions:

- <u>**Initial**, where the amount equals 0 – MIT establish:</u>

  With this transaction, the merchant obtains consent to initiate further transactions without the customer being present. It is a Merchant Initiated Transaction (MIT) where card details need to be entered or the card is already stored with a Register transaction from before. In both cases the fields are filled in the same way.

  - In the "*transaction Type*" field, the transaction is marked PREAUTHORIZE.
  - In the "*transactionIndicator*" field, the transaction needs to be marked INITIAL, as it is the first transaction in the series.
  - The "withRegister" field should be marked TRUE, which means that the card is stored for subsequent payments.
  - The "*authenticationIndicator*" field is automatically set to 02 for this type of transaction, as it is linked to the "*transaction Type*" field. The value "02" means that the card is also stored when the MIT series is created.
  - The field "recurringFrequency" is populated by default with the value "1" because, according to the strict rules of the card schemes, it is not a recurring transaction, as the amount can be different each month for MIT transactions. The desired number of payments in the series can be entered here.
  - If the transaction is one where the card is already stored, the "*referenceTransactionId*" field should be used and populated with the UUID (unique indicator) of the Register or the initial transaction in the series, as the system thus knows that strong authentication has already been performed on the first transaction and does not need to be performed here.
  - The field "*challengeIndicator"* is populated with the value "04", which means that strong authentication (SCA) is required by the card schemes for each initial transaction.

- <u>**Initial**, where the amount is different from 0 – MIT establish:</u>

  It is a Merchant Initiated Transaction (MIT) where card details either need to be entered or the card is already stored with a Register transaction from before.

  In both cases the following fields are filled in the same way.

  - This transaction sets up a series of MIT transactions and actually purchases goods at the same time, so the "*transaction Type*" field is marked as DEBIT or PREAUTHORIZE, depending on which transaction type the merchant supports or wants to perform.

For the other fields the same rules apply as for the initial transaction where the amount is 0.

- **<u>Subsequent MIT</u>** <u>transactions</u>:
  It is a Merchant Initiated Transaction (MIT) using a previously stored card.
  - The "*transaction Type*" field is marked DEBIT because it is a single-phase transaction.
  - In the "*transaction Indicator*" field, the transaction should be marked as CARDONFILE-MIT, thus indicating that it is a subsequent transaction, but linked to the first one.
    - Therefore, the field "*referenceTransactionId*" should be filled with the UUID (unique indicator) of the Initial transaction, because the system thus knows that strong authentication has already been performed on the first transaction and does not need to be performed here.
  - Other fields marked above do not need to be filled in.

Example of a code for INITIAL (first) transactions, in JSON format on the next page:

```
{
  "merchantTransactionId": "D-2022-04-07-624eb9146f6bf",
  "extraData": {
    "userField1": "00",
   }
   "threeDSecureData":{
    "authenticationIndicator": "02",
    "recurringFrequency": "1"
  },
  "merchantMetaData": "Transaction:Debit;Description:test",
  "amount": "1.99",
  "currency": "EUR",
  "successUrl": https://mydomain.com/PHPPaymentGatewayJson/examples/PaymentOK.php",
  "cancelUrl": " https://mydomain.com/PHPPaymentGatewayJson/examples/PaymentCancel.php",
  "errorUrl": " https://mydomain.com/PHPPaymentGatewayJson/examples/PaymentNOK.php",
  "callbackUrl": "https://mydomain.com/PHPPaymentGatewayJson/examples/Callback.php",
  "description": "One pair of shoes",
  "withRegister": true,
  "transactionIndicator": "INITIAL",
  "customer": {
    "firstName": "Janez",
    "lastName": "Novak",
    "billingAddress1": "Street 1",
    "billingCity": "City",
    "billingPostcode": "1000",
    "billingCountry": "SI",
    "email": "test@email.com",
    "ipAddress": "91.208.168.48"
  },
  "language": "sl"
}
```

Example of a code for SUBSEQUENT (further) transactions, in JSON format:

```
{
  "merchantTransactionId": "D-2022-04-07-624ebdc06bed2",
  "extraData": {
    "userField1": "00"
  },
  "merchantMetaData": "Transaction:Debit;Description:test",
  "referenceUuid": "2a06c2a09b69c8eced14",
  "amount": "1.99",
  "currency": "EUR",
  "successUrl": https://mydomain.com/PHPPaymentGatewayJson/examples/PaymentOK.php",
  "cancelUrl": " https://mydomain.com/PHPPaymentGatewayJson/examples/PaymentCancel.php",
  "errorUrl": " https://mydomain.com/PHPPaymentGatewayJson/examples/PaymentNOK.php",
  "callbackUrl": "https://mydomain.com/PHPPaymentGatewayJson/examples/Callback.php",
  "description": "One pair of shoes",
  "withRegister": false,
  "transactionIndicator": "CARDONFILE-MERCHANT-INITIATED",
  "customer": {
    "firstName": "Janez",
    "lastName": "Novak",
    "billingAddress1": "Street 1",
    "billingCity": "City",
    "billingPostcode": "1000",
    "billingCountry": "SI",
    "email": "test@email.com",
    "ipAddress": "91.208.168.48"
  },
  "language": "sl"
}
```

3.5.3.3. Indicating Recurring transactions:

JSON	INITIAL - AMT 0		INITIAL		SUBSEQUENT
Type	**RECURRING** establish		**RECURRING** establish		**RECURRING**
Card number	enter	stored	enter	stored	stored
transaction Type	PREAUTHORIZE		PREAUTHORIZE or DEBIT		DEBIT
transactionIndicator	INITIAL		INITIAL		RECURRING
referenceTransactionId(referenceUuid)	\	UUID(register)	\	UUID(regi	UUID(initial)
withRegister	TRUE	TRUE	TRUE	TRUE	\
authenticationIndicator	02	02	02	02	\
recurringFrequency	<>1	<>1	<>1	<>1	\
challengeIndicator	04	04	04	04	\
authenticationIndicator, recurringFrequency in challengeIndicator se pošljejo v ThreeDSecureData					

Figure 7: Table with indicators for Recurring transactions

Within Recurring transactions we again have 3 types of possible implementations of these transactions:

- **Initial**, where the amount equals 0 – Recurring establish:

  This transaction sets up a series of MIT transactions. It is a Merchant Initiated Transaction (MIT) where card details need to be entered or the card is already stored with a Register transaction from before.

  In both cases the following fields are filled in the same way.

  - In the "*transaction Type*" field, the transaction is marked PREAUTHORIZE.
  - In the "*transactionIndicator*" field, the transaction needs to be marked INITIAL, as it is the first transaction in the series.
  - The "withRegister" field is marked TRUE, which means that the card is kept stored.
  - The "*authenticationIndicator*" field is automatically set to 02 for this type of transaction, as it is linked to the "*transaction Type*" field. The value "02" means that the card is also stored when the MIT series is created.

  - The "recurringFrequency" field is populated by default with a value different from 1 or with the number of transactions to be executed in this series. The desired number of payments in the series can be entered here.
  - If the transaction is one where the card is already stored, the "*referenceTransactionId*" field should be used and populated with the UUID (unique indicator) of the Register or the initial transaction in the series, as the system thus knows that strong authentication has already been performed on the first transaction and does not need to be performed here.
  - The field "*challengeIndicator"* is populated with the value "04", which means that strong authentication (SCA) is required by the card schemes for each initial transaction.

- **<u>Initial</u>**<u>, where the amount is different from 0 – Recurring establish:</u>

  It is a Merchant Initiated Transaction (MIT) where card details either need to be entered or the card is already stored with a Register transaction from before. In both cases the following fields are filled in the same way.
  - This transaction sets up a series of MIT transactions and actually purchases goods at the same time, so the "*transaction Type*" field is marked as DEBIT or PREAUTHORIZE, depending on which transaction type the merchant supports or wants to perform.

  For the other fields the same rules apply as for the initial transaction where the amount is 0.

- **<u>Subsequent Recurring</u>** <u>transactions:</u>

  It is a Merchant Initiated Transaction (MIT) using a previously stored card.
  - The "*transaction Type*" field is marked DEBIT because it is a single-phase transaction.
  - In the "*transaction Indicator*" field, the transaction should be marked as RECURRING, thus indicating that it is a subsequent transaction, but linked to the first one.
    - Therefore, the field "*referenceTransactionId*" should be filled with the UUID (unique indicator) of the Initial transaction, because the system thus knows that strong authentication has already been performed on the first transaction and does not need to be performed here.
  - Other fields marked above do not need to be filled in.

Example of a code for INITIAL (first) transactions, in JSON format (next page):

```
{
   "merchantTransactionId": "D-2022-04-07-624eb9951e2ac",
   "extraData": {
      "userField1": "05",
    "threeDSecureData":{
      "authenticationIndicator": "02",
      "recurringFrequency": "2"
   },
   "merchantMetaData": "Transaction:Debit;Description:test",
   "amount": "1.99",
   "currency": "EUR",
   "successUrl": https://mydomain.com/PHPPaymentGatewayJson/examples/PaymentOK.php",
   "cancelUrl": " https://mydomain.com/PHPPaymentGatewayJson/examples/PaymentCancel.php",
   "errorUrl": " https://mydomain.com/PHPPaymentGatewayJson/examples/PaymentNOK.php",
   "callbackUrl": "https://mydomain.com/PHPPaymentGatewayJson/examples/Callback.php",
   "description": "One pair of shoes",
   "withRegister": true,
   "transactionIndicator": "INITIAL",
   "customer": {
      "firstName": "Janez",
      "lastName": "Novak",
      "billingAddress1": "Street 1",
      "billingCity": "City",
      "billingPostcode": "1000",
      "billingCountry": "SI",
      "email": "test@email.com",
      "ipAddress": "91.208.168.48"
   },
   "language": "sl"
}
```

Example of a code for SUBSEQUENT (further) transactions, in JSON format:

```
{
   "merchantTransactionId": "D-2022-04-07-624ebf19af16d",
   "extraData": {
      "userField1": "00"
   },
   "merchantMetaData": "Transaction:Debit;Description:test",
   "referenceUuid": "4a58b83ad5794fa66041",
   "amount": "1.99",
   "currency": "EUR",
   "successUrl": https://mydomain.com/PHPPaymentGatewayJson/examples/PaymentOK.php",
   "cancelUrl": " https://mydomain.com/PHPPaymentGatewayJson/examples/PaymentCancel.php",
   "errorUrl": " https://mydomain.com/PHPPaymentGatewayJson/examples/PaymentNOK.php",
   "callbackUrl": "https://mydomain.com/PHPPaymentGatewayJson/examples/Callback.php",
   "description": "One pair of shoes",
   "withRegister": false,
   "transactionIndicator": "RECURRING",
   "customer": {
      "firstName": "Janez",
      "lastName": "Novak",
      "billingAddress1": "Street 1",
      "billingCity": "City",
      "billingPostcode": "1000",
      "billingCountry": "SI",
      "email": "test@email.com",
      "ipAddress": "91.208.168.48"
   },
   "language": "sl"
```

3.6. Support for online instalment payments

As part of the online payment support, it is also possible to introduce instalment payments. This service is available to merchants who have a contract with a bank that specifies this service in its offer.
The contract includes support on the Bankart systems side (Payment Gateway, authorization system, posting data preparation system), but at the same time support must be put in place on the merchant's side to offer this service as a whole to the customer. Below is a description of the support and how to implement the support on the merchant's side.
The basis for processing instalments is that the merchant sends the desired number of instalments in which payment will be made by the buyer in the API call. Upon purchase, the authorization to the card will be made based on the total amount of the purchase.

The process of introducing online instalment payment support is as follows:

- The merchant enters into an agreement with the bank to introduce online instalment payment support.
- The bank communicates to Bankart the merchant code of the merchant with whom the instalment payment contract is concluded.
  Based on this information, the instalment payment option is opened in the authorization system. In addition to the authorization to carry out instalment payments, the following controls are made as part of the authorization system control:
    - The number of instalments is specified in the contract concluded with the bank. If a request is sent with a value of 0 or 1 in the field for the number of instalments, the purchase is considered as a purchase without instalments.
    - If unauthorized characters are received (only numeric characters are allowed), the transaction is rejected.
    - If the merchant does not have the option to allow instalment payments selected, the value in the received instalment field is ignored by the merchant and the transaction is executed without instalments.
- An existing field in Payment Gateway called "*userField1*" is used to forward instalments to the authorization system (*visible in the above examples of the JSON call for card storing*). Until now, this field has not been used (the value entered had no impact on the processing of transactions). If the merchant has a signed instalment payment support contract and a solution in place, the value in the "*userField1*" field has an impact on the processing.
- The number of instalments is entered by the user. To avoid problems with inappropriate data entry by the customer, we recommend the use of drop-down menus, radio buttons, etc., and thus limiting the possible values entered to only the number specified in the contract concluded with the bank. The merchant may introduce additional instalment controls on their site and, after the customer has entered the number of instalments they wish to make in relation to the items purchased or the amount, either allow such a transaction (forward it for further processing) or not allow the customer's choice (e.g. prevent a purchase of EUR 10 in 5 instalments, which would incur excessive costs in terms of bank charges).
- In the process of displaying the possible payment methods, the merchant must give the customer the choice of a method that covers instalment payments with the cards for which the merchant has a contract in place. If the customer selects the instalment payment method, the merchant enables the customer to select or (less preferably) enter the desired number of instalments. The number of instalments selected/entered by the customer is sent by the merchant in the request in the "*userField1*" field.
- Once the request has been forwarded, it will be handled in line with the regular production procedures with the addition of an instalment field control. Authorization is made for the full

amount of the purchase.

- The merchant carries out a CAPTURE in accordance with the dynamics of the dispatch of the goods, whether or not payment by instalments has been selected.
- If the merchant carries out CAPTURE of the purchase in several parts, each part will be deemed charged in line with the number of instalments selected.
- If the merchant has instalment payment support in place (based on a contract with the bank) and also provides the number of instalments in the request, the merchant may consider, in the event of a positive response, that the customer will be debited by the bank within the amounts of the individual instalments. In such a case, the merchant should note in the receipt to the customer that the purchase has been made in instalments, in the number sent in the request.
- The number of instalments is visible to the merchant in the instalment display field of the online store (depending on the way the developer has implemented it) and on the website https://tebank.bankart.si/Testgw31/Merchant/index.jsp under the Transactions menu.
- The bank charges for instalment transactions are covered as agreed with the bank,
- The merchant implements the installment selection menu themselves, as support for this is not available on the HPP entry mask.

4. TESTING GUIDELINES

4.1. Basic information

Before moving to a production environment for the use of online payments, we at Bankart require the successful completion of a few tests to confirm the correct functioning of the merchant's system and connection to Bankart's Payment Gateway . Before starting the testing, the Bankart customer support team (customer.support@bankart.si) will provide you with some the credentials to connect to the Bankart Payment Gateway and start the testing. Here are the main credentials and in the next subsection you will be shown how to obtain all of them.

Name	Description
API username	API access – the developer uses this in the code.
API password	Access to API
API Key	Payment method key.
Shared Secret	Shared secret for signing and authenticating the content of API calls.
Public Integration Key (PKI)	Key to integrate payment.jst element in your store.
Web access:	
URL	Web address for accessing the portal and viewing transactions (refers to the following section)
Username	Username assigned for accessing the portal
Password	Password assigned for first login (to be changed after first login!)

4.2. Obtaining data for connection to Payment Gateway

The data to enter the Payment Gateway portal is sent via two channels:

- the **email address** provided at your registration at the bank (This is in use only for Slovenian merchants). For merchants that are part of NLB Skopje, NLB Sarajevo, NLB Banja Luka, NLB Podgorica, NLB Priština, we will send two emails to the banks' department for e-commerce, who will then forward this information to you the merchant. In the first email we will send the integration instructions and the password to access the portal, which must be changed immediately upon login. And in the second email we will send you the username to access the portal.
- **Text message** to the number provided upon merchant registration. The username to access the portal will be sent via this channel. (This is in use only for Slovenian merchants)

Once you have received the login details for the portal, you will then be able to collect the terminal connection/integration details, as shown in the step-by-step instructions below:

Step 1: After logging in to the Payment Gateway portal, select the "Connectors" tab (*highlighted in green*) on the left.

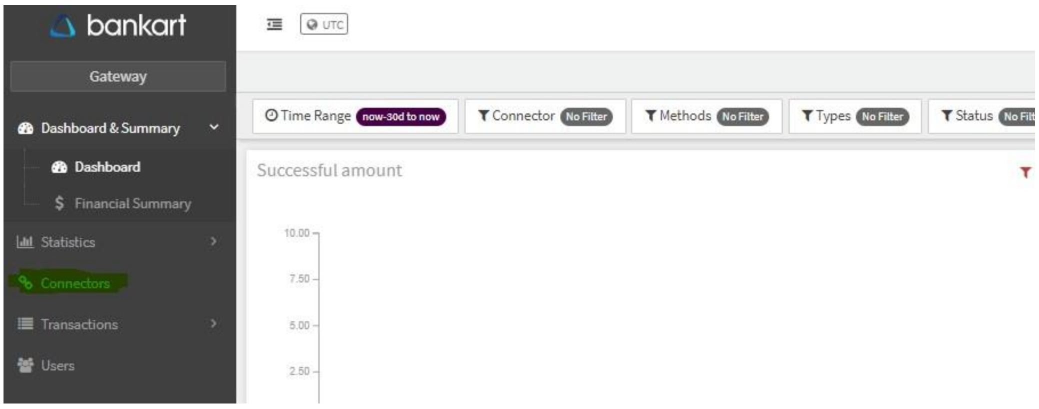
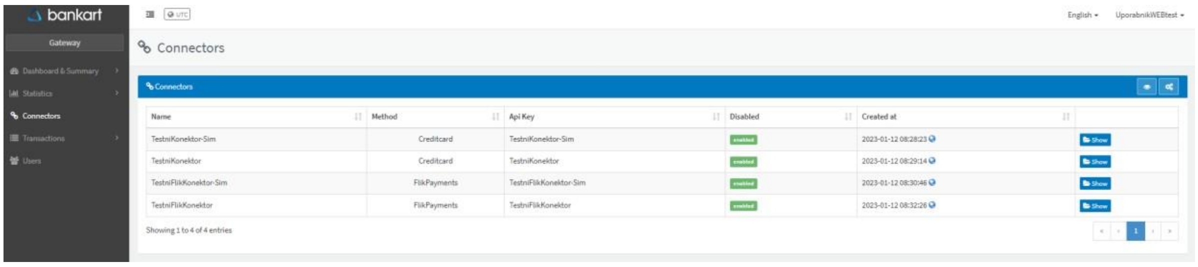


Figure 8: Selecting the "Connectors" tab in Payment Gateway

Step 2: A list of all terminals or connectors is displayed. For each of the connectors, click on the

"Show" button on the right to access the information relevant for the integration or connection to Payment Gateway (in the next step). Connectors that have "Sim" at the end are simulation connectors and are used to test the integration before the merchant goes into production.
More about the actual testing with simulation connectors is provided later in the document.



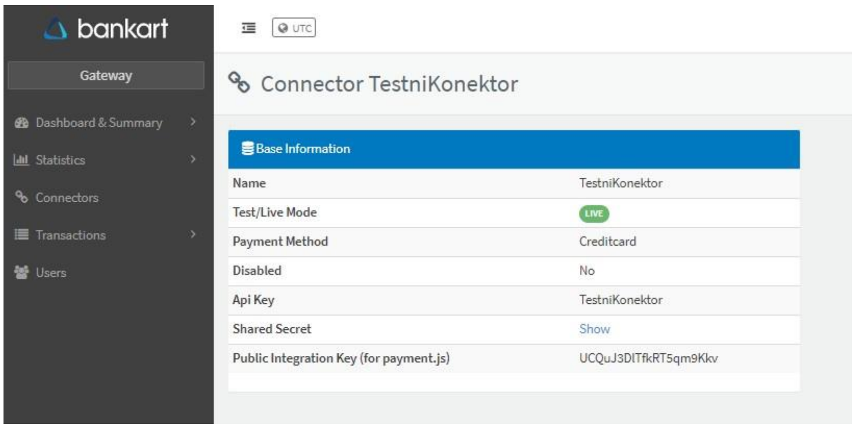
Name	Method	Api Key	Disabled	Created at	
TestniKonektor-Sim	Creditcard	TestniKonektor-Sim	enabled	2023-01-12 08:28:23	Show
TestniKonektor	Creditcard	TestniKonektor	enabled	2023-01-12 08:29:14	Show
TestniFlikKonektor-Sim	FlikPayments	TestniFlikKonektor-Sim	enabled	2023-01-12 08:30:46	Show
TestniFlikKonektor	FlikPayments	TestniFlikKonektor	enabled	2023-01-12 08:32:26	Show

Showing 1 to 4 of 4 entries

Figure 9: Display of connectors in Payment Gateway

Step 3: Click on the "Show" button to see more detailed information about the connector. The merchant requires the following information to connect to Gateway:

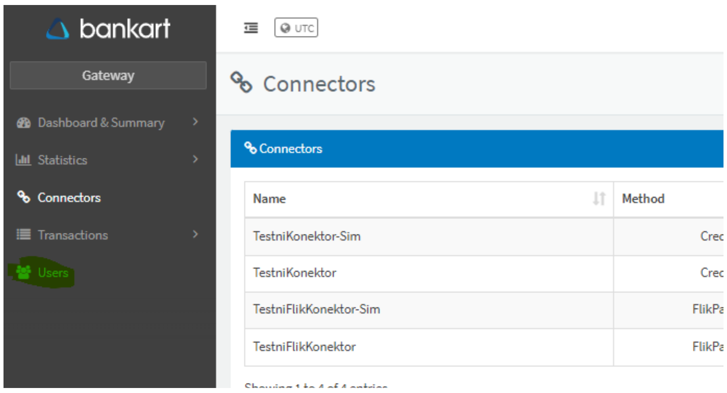
a. API Key
b. Shared Secret (click on "show" to display the information)
c. Public Integration Key (needed in case of *payment.js* integration for card payments – if the merchant doesn't want to redirect to Bankart's input mask, but wants an implementation on their own site to make it look like the customer never leaves the site/store.)



Name	TestniKonektor
Test/Live Mode	LIVE
Payment Method	Creditcard
Disabled	No
Api Key	TestniKonektor
Shared Secret	Show
Public Integration Key (for payment.js)	UCQuJ3DITfkRT5qm9Kkv

Figure 10: Display of connector data that needs to be copied into the merchant's systems

Step 4: After the data for each of the desired connectors has been copied into the merchant's system, select the "Users" tab (*highlighted in green*) from the menu on the left.

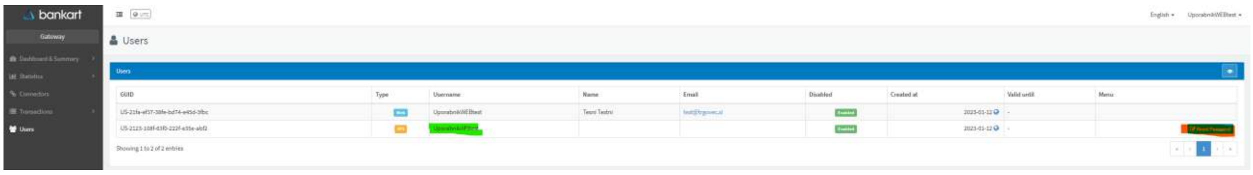


Name	Method
TestniKonektor-Sim	Crec
TestniKonektor	Crec
TestniFlikKonektor-Sim	FlikPa
TestniFlikKonektor	FlikPa

Figure 11: Display of the selection of the "Users" tab in Payment Gateway

Step 5: A list of all the merchant's users will be displayed. There can be more than one WEB type user, as these are the users for whom the merchant enables access to the Payment Gateway portal and transaction review (we suggest up to a maximum of 3, created by Bankart). There is usually only one API type user, as this is the one through which the connection to Payment Gateway is enabled.

The data for the main WEB user has already been sent to the merchant. The merchant has already taken over the connector data. Finally, the API user data still needs to be obtained. For this part, the merchant has to copy the API username (*highlighted in green*) to their system, and for this same API user the "Reset Password" button (*highlighted in red*) has to be clicked in Payment Gateway on the right side of the page.



GUID	Type	Username	Name	Email	Disabled	Created at	Valid until	Menu
US-21fa-ef37-38fe-bd74-e45d-3fbc	Web	UporabnikWEBtest	Testni Testni	test@trgovec.si	Enabled	2023-01-12	-	
US-2123-108f-63f0-222f-e35e-abf2	API	UporabnikAPItest			Enabled	2023-01-12	-	Reset Password

Showing 1 to 2 of 2 entries

Figure 12: Displaying users in Payment Gateway with a focus on API users

Step 6: The menu for generating the API password will appear. Clicking on the "Generate" option (*highlighted in green*) under the Password field will generate the password. You have to **copy** this password **into the merchant's systems**, then click on the "Reset Password" button (*highlighted in yellow*) on the bottom right. This saves the password in the Payment Gateway system.

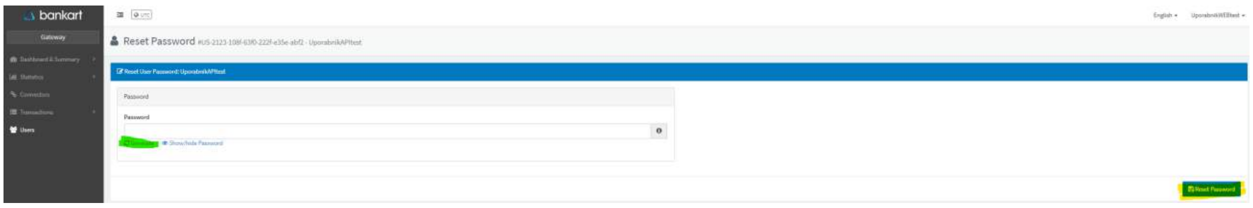
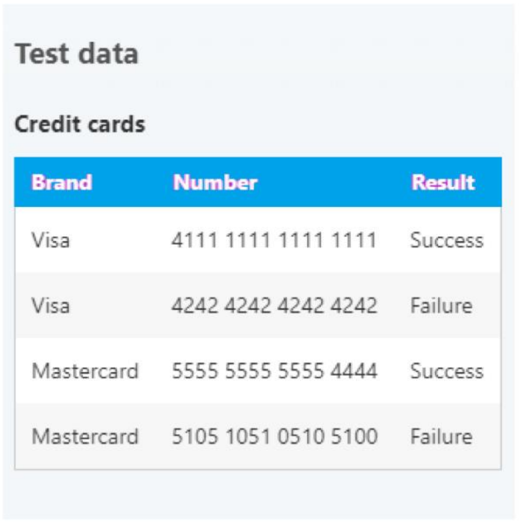


Figure 13: Display of a password change for an API user

IMPORTANT: The API password generation should be done **only once or at the first login**. Before the password generation process is completed, the password should be saved and copied to the merchant's back-office systems so that the merchant can connect to Payment Gateway. If the API password is changed after the merchant has been in production for a long time and uses the API password generated upon initial login, the connection between the merchant and Payment Gateway will **BREAK** after a subsequent password change and transactions **will NOT be able to** be run and executed correctly.

4.3. Test cards

Testing must be done with the test cards listed on the right. In the event of unexpected responses, please contact customer.support@bankart.si.
Enclosed are the details of the cards that can be used to test the correct functioning of the transactions for each card scheme.
First, **the test card numbers for VISA and Mastercard** (see figure on the right). Depending on the desired result (which is written next to the card number), select one of the card numbers. In the field where the CVC code is checked, you can enter any three-digit combination, and the same applies to the validity date, where any (valid) date can be entered.



Test data

Credit cards

Brand	Number	Result
Visa	4111 1111 1111 1111	Success
Visa	4242 4242 4242 4242	Failure
Mastercard	5555 5555 5555 4444	Success
Mastercard	5105 1051 0510 5100	Failure

Figure 14: VISA and Mastercard test cards

Below are the **Diners test card** numbers in case you have a contract with Diners or Sparkasse Pay Slovenia:
3800 0000 0000 06 – *the expected response is* ***transaction approved***
3614 8900 6479 13 – *the expected response is* ***transaction rejected***

As above, select one of the card numbers according to the desired result if the integration is correct. You can enter any three-digit combination in the field where the CVC code is checked, and the same applies to the validity date, where any (valid) date can be entered.
Test cards are for testing purposes only and may NOT be used in production.
Production cards issued by banks are **NOT** to be used for testing in the test system, but only test cards agreed with Bankart.

4.4. Testing with the simulation connector

For testing purposes, we will define your test merchant to verify the functioning of your system. The response will be simulated on the payer's side.
After you initiate the transaction, an input mask will be displayed (shown in the figure) if you are using the HPP redirect, or an integrated input mask will be displayed in your online store if you are using the payment.js integration. In both cases, you need to enter the test card details provided to you by Bankart or your bank.

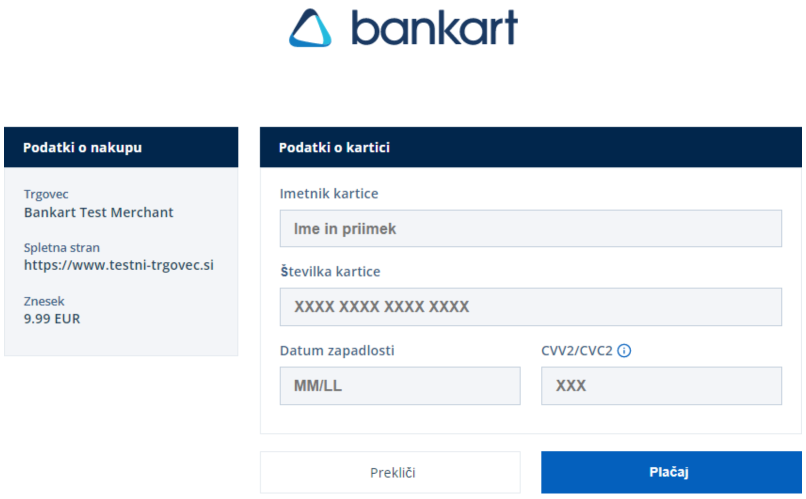


Figure 15: Bankart's new input mask

Podatki o nakupu	Purchase information
Trgovec Bankart Test Merchant	Merchant Bankart Test Merchant
Spletna stran https://www.testni-trgovec.si	Website https://www.testni-trgovec.si
Znesek 9,99 EUR	Amount EUR 9.99
Podatki o kartici	Card details
Ime in priimek	Name and surname
Številka kartice XXXX XXXX XXXX XXXX	Card number XXXX XXXX XXXX XXXX
Datum zapadlosti MM/LL	Expiry date MM/YYYY
CVV2/CVC2 XXX	CVV2/CVC2 XXX
Prekliči	Cancel
Plačaj	Pay

After entering your details and clicking the "Pay" button, you will be presented with a 3D Secure authentication simulation screen in the test environment (not in production) and the option to select the authentication result. The card schemes use an ECI value of 05 as an indicator that a strong authentication (SCA) has been performed, so in most cases you will either select this option or the "Failed" option.

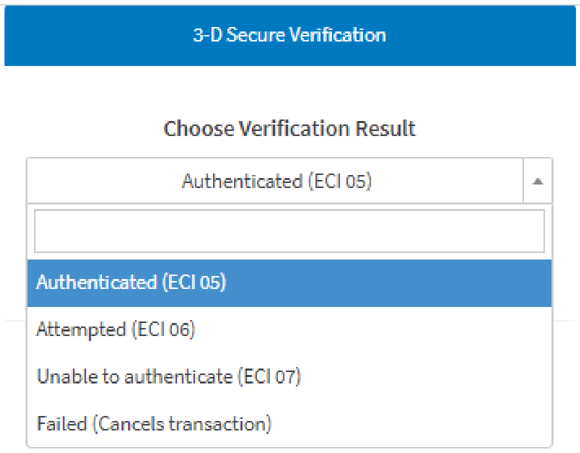

3-D Secure Verification
Choose Verification Result
Authenticated (ECI 05)
Authenticated (ECI 05)
Attempted (ECI 06)
Unable to authenticate (ECI 07)
Failed (Cancels transaction)

Figure 16: Choice of authentication result for the simulation connector

- **ECI 05:** Transaction is authenticated
- **ECI 06:** Failed authentication attempt in the transaction
- **ECI 07:** Transaction was not authenticated
- **Failed:** Authentication is not performed, transaction is terminated

After clicking on the submit button, the transaction is executed to the end, where a screen with the transaction success result is then displayed.

4.5. Transition to a production environment

- Bankart's customer support checks the integration status via the Payment Gateway logs:
  Before granting production access, our customer support checks the Payment Gateway logs to see if the payment flows are correctly executed on the simulation adapter. If everything is in order or the transactions are correctly executed and marked, Bankart's customer support team will manually enable the option to use the production terminal.
- Technical support during working hours only:
  An important reminder that we always send out in our merchant onboarding emails is that technical support is only available during working hours (08:00–15:00), at the customer.support@bankart.si email, on business days, as freelance developers may also work outside regular working hours or over the weekend.
- Merchants should **NOT** conduct testing in a production environment!
  It is important that merchants do not make changes to the production instance of their connection to Payment Gateway. To this end, we provide merchants with a test environment in which to confirm the functioning and then enable the same configuration in the production environment. In this way we ensure that the moment a merchant receives production access, their integration with Payment Gateway is working. If changes are subsequently made to the production environment WITHOUT further testing, the merchant assumes responsibility for any problems in the execution of individual transactions. If the merchant makes any significant changes to the settings in the production environment, particularly outside of working hours, we cannot provide you with immediate technical assistance.

4.6. Procedure for Action in Case of Transaction Denial with Specific Reason

In certain cases, the merchant will receive information about the reason for the transaction denial in the Postback request sent to the Callback URL. This information will be included in the field below, where "x" represents the number/code sent:

```
"extraData": {
        "psp:fn.respCdeCat": "x"
}
```

Based on the value in the "*psp:fn.respCdeCat*" field, within the "*extraData*" object, the merchant is required to take action in certain cases. This includes notifying the customer that no further transactions can be made on their side for a specified period, or even temporarily blocking the customer from making new transactions using that card.

Below are the codes ("*x*" in the example above), their explanations, and the appropriate actions the merchant must take in each case:

- "**psp:fn.respCdeCat**":"**1**"
  The transaction with the entered card details was not approved, which means one or more incorrect details were entered. The customer must re-enter the correct details for the transaction to be successfully completed.
- "**psp:fn.respCdeCat**":"**2**"
  The transaction with the entered card details will not be approved at the moment. It is recommended that the customer try again later. In certain cases, depending on the status of the

card, two additional fields may appear in the "extraData" object, specifying the time frame for retrying the transaction:

- First, the "**psp:fn.retryTim**" field, which contains a value from 0 to 99 and indicates the time in minutes, hours, or days that must be waited before attempting the transaction again.
- And the "**psp:fn.retryPrd**" field, which indicates the unit of time related to the previous field:
    - "0" means minutes,
    - "1" means hours,
    - "2" means days.

- "**psp:fn.respCdeCat**":"**3**"
  The transaction with the entered card details will never be approved because the card is blocked by the card scheme. The customer should not attempt to make any further transactions with this card.

For the above cases, the merchant must prepare appropriate error status pages that will contain notifications clearly reflecting the meaning of the individual denial statuses.
It is important to emphasize that these denial codes will not appear for every transaction but only in cases where the transaction was made with a card from a foreign bank.

5. BANKART PAYMENT GATEWAY: USER INTERFACE OVERVIEW

5.1. Access to the portal

The web interface is available at: https://gateway.bankart.si/en/login
The username and password for online access are separate from the API access. You will also provide an email address when you arrange formalities with the bank, which will also be used in case you need to reset your password.
You should handle any changes to your details with your bank.
If you have any problems logging in, please contact customer.support@bankart.si for assistance. You can also contact Customer Support if you are unclear about or have any problems using the interface.

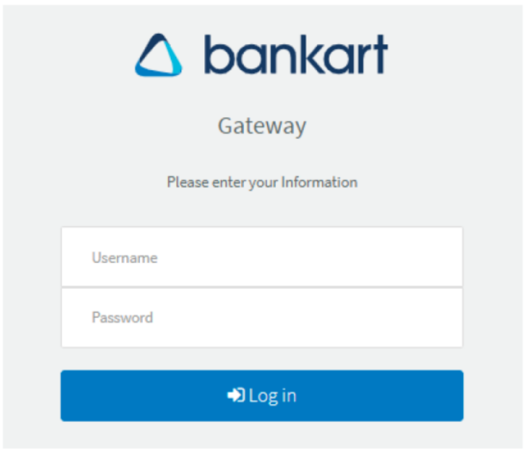


Figure 17: Entry point for the Payment Gateway portal

5.2. Overview of the graphical interface

After entering the data and signing in, the home page or dashboard of the portal appears.
On the left, in the dark grey area, there are different tabs where you can choose to display the desired details. The most useful tabs for you are **Dashboard & Summary** and **Transactions**.

After a successful login, the first tab of the graphical interface, “Dashboard”, is displayed, where you can see an overview of the statistics of your transactions. Here you can adjust the filters/parameters by clicking on the top right “Customize” button (*highlighted in red*) and display the desired graphs and figures according to your preferences.
When you log in for the first time, please also select the time zone set to Europe/Ljubljana (*highlighted in yellow*) on the top left to ensure that the transaction times are displayed in the correct time and that there are no problems when searching for transactions.

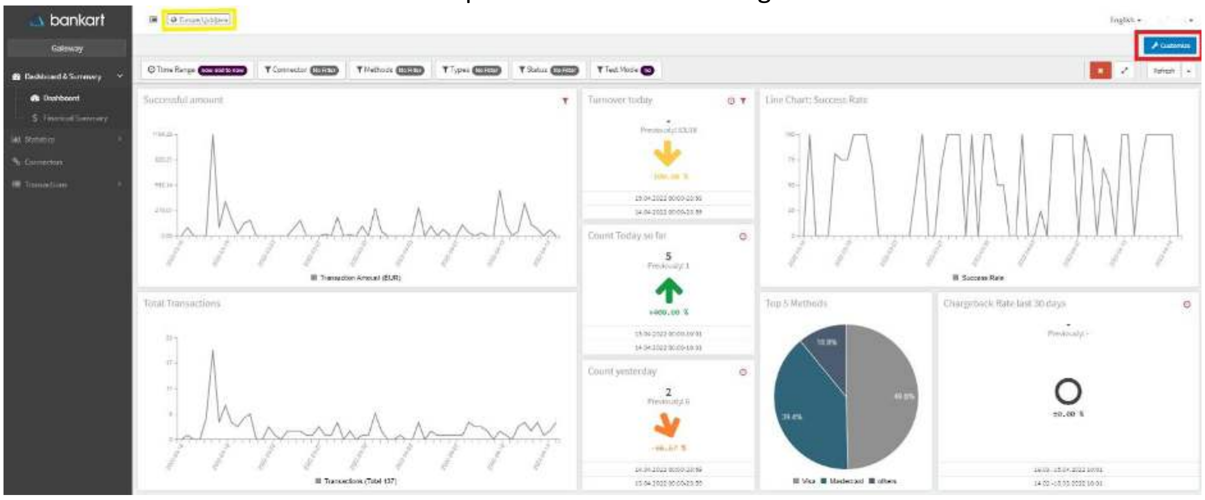
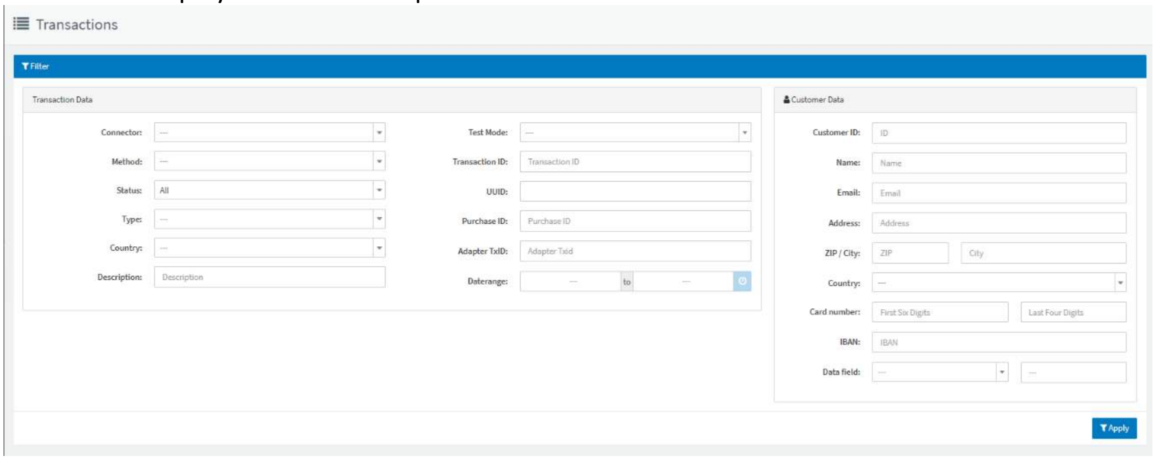


Figure 18: First page in the Payment Gateway portal

On the left, in the grey multi-tab menu, you can select the different functions offered by the user interface. As mentioned above, you will probably find the “Transactions” tab the most useful, where you can access a list and information about the transactions activated through your online store.

In the upper part of the page you can use various parameters to search for a specific transaction, which is then displayed in the lower part:



Transactions

Filter

Transaction Data

Connector:	---	Test Mode:	---
Method:	---	Transaction ID:	Transaction ID
Status:	All	UUID:	
Type:	---	Purchase ID:	Purchase ID
Country:	---	Adapter TxID:	Adapter TxId
Description:	Description	Daterange:	--- to ---

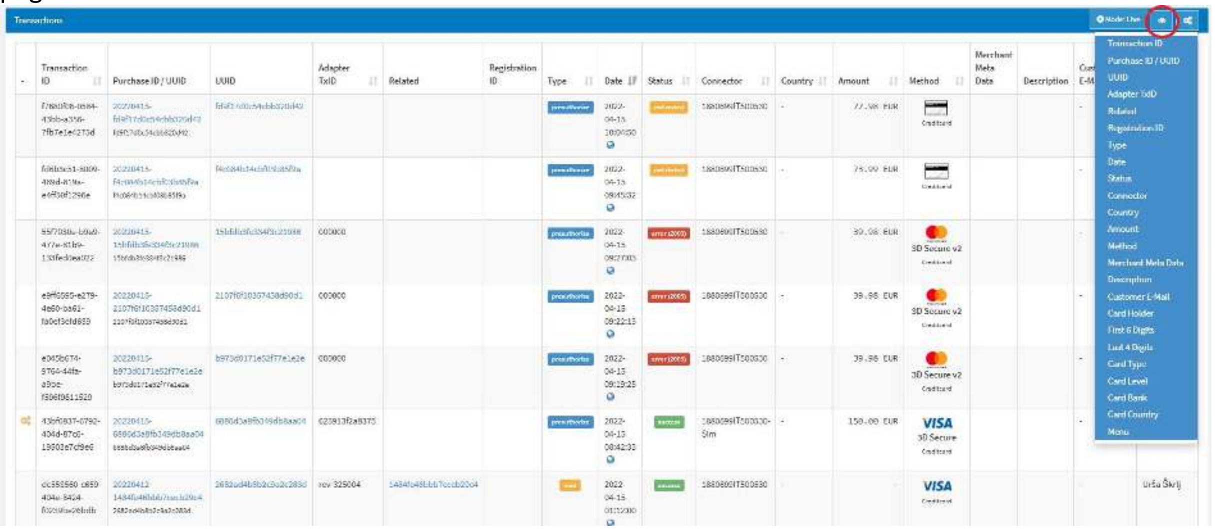
Customer Data

Customer ID:	ID
Name:	Name
Email:	Email
Address:	Address
ZIP / City:	ZIP / City
Country:	---
Card number:	First Six Digits / Last Four Digits
IBAN:	IBAN
Data field:	--- / ---

Apply

Figure 19: Display of filters for searching transactions in Payment Gateway

The bottom part basically shows a list of all the transactions activated, where the transactions are sorted in terms of time, from the newer transactions activated to the older ones. In the figure below, two buttons can be seen on the top right of the blue bar. Clicking on the left button (*highlighted in red in the figure below*) will bring up a sub-menu where you can adjust the visibility of the individual fields on the list according to your needs. If the field name is coloured blue, then this field is displayed, and if it is coloured grey it is not displayed. Alternatively, you can right-click to quickly export a single page of transactions in the *.csv* format.



Transactions

-	Transaction ID	Purchase ID / UUID	UUID	Adapter TxID	Related	Registration ID	Type	Date	Status	Connector	Country	Amount	Method	Merchant Meta Data	Description	Customer E-Mail
	f78a0f08-0684-43bb-a356-7fb7e1e62T3d	20220415-fd9f17d0e54ebb020d42	fd9f17d0e54ebb020d42				[illegible]	2022-04-15 10:04:50	[illegible]	1880896IT500630	-	77.98 EUR	Creditcard			-
	fd6b5e51-8009-488d-819a-e6ff30f1296e	20220415-f4c084b14c5f05d85f9a	f4c084b14c5f05d85f9a				[illegible]	2022-04-15 08:45:32	[illegible]	1880896IT500630	-	75.00 EUR	Creditcard			-
	55f7038a-b9a0-477e-81b9-133fed0ea022	20220415-15bfdb3fc3348fc21986	15bfdb3fc3348fc21986	C00000			preauthorize	2022-04-15 09:27:05	error (2003)	1880690IT500630	-	39.98 EUR	3D Secure v2 Creditcard			-
	e9ff6595-e279-4e60-ba61-fa0ef3efd659	20220415-2107f6f10307458d90d1	2107f6f10307458d90d1	C00000			preauthorize	2022-04-15 09:22:15	error (2005)	1880699IT500630	-	39.98 EUR	3D Secure v2 Creditcard			-
	e045b074-9764-44fa-a9be-f9069611520	20220415-b973d0171e52f77e1e2e	b973d0171e52f77e1e2e	C00000			preauthorize	2022-04-15 09:19:25	error (2005)	1880699IT500630	-	39.98 EUR	3D Secure v2 Creditcard			-
	43bf0837-0792-404d-87c0-19902e7cf9e6	20220415-6880d3a8fb349db8aa04	6880d3a8fb349db8aa04	623913f2a8375			preauthorize	2022-04-15 08:42:33	success	1880699IT500630-Sim	-	150.00 EUR	VISA 3D Secure Creditcard			-
	dc558560-c659-404e-8424-f0239[illegible]	20220412-1484fb48bbb7[illegible]	2682ad4b5b2c5a2c283d	rev 325004	1484fb48bbb7cccb2004		[illegible]	2022-04-15 01:12:00	[illegible]	1880690IT500630			VISA Creditcard			Urša Škrlj

Figure 20: Option to select the display of fields in the “Transactions” tab

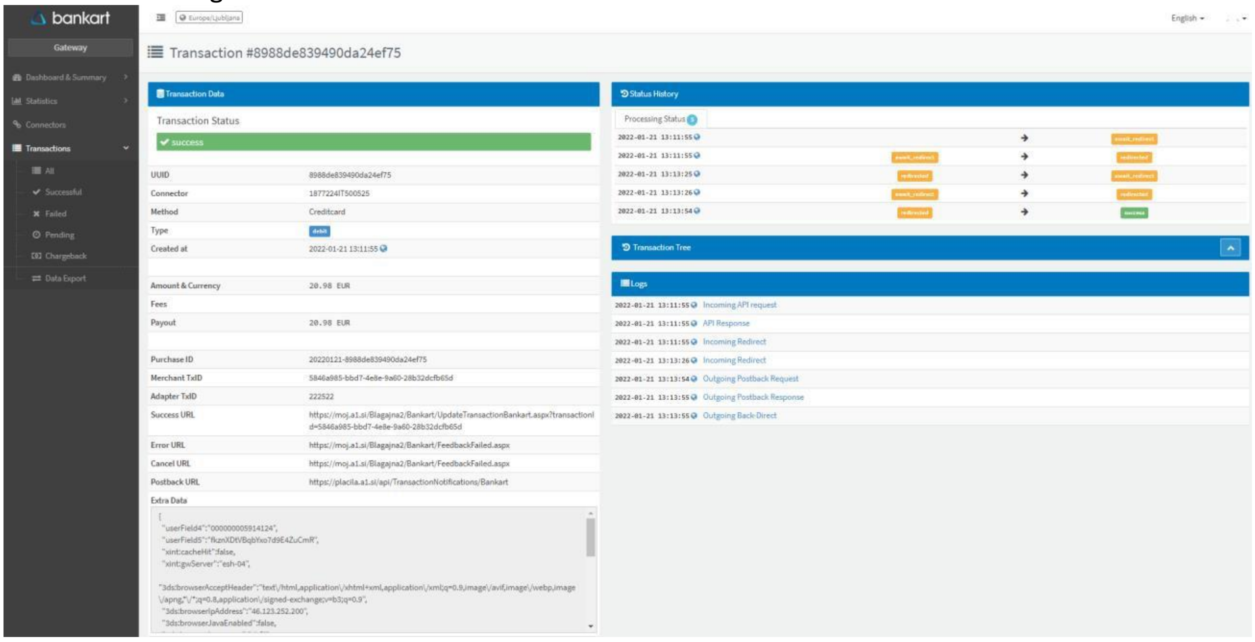
The most important fields on the list of transactions are:

- ***Transaction ID***: the transaction identifier, set by the merchant and fed into the system, which must be unique for each transaction. The IDs must be different for preauthorization and subsequent capture or void.
- ***Purchase ID/UUID:*** under this value you can pair preauthorization with capture or void. The UUID is simply a unique transaction ID set by Payment Gateway itself.
- ***Type:*** type of transaction activated (debit, preauthorization, capture, void, etc.)
- ***Status:*** transaction success

- ***Amount:*** transaction amount
- ***Method:*** card scheme and user authentication version

5.2.1. Reviewing the details (logs) of each transaction

In the list of transactions, clicking on the "Details" button (located at the far right of each transaction on the list), or clicking on the number in the UUID field, will bring up an interface where you can view the details and logs of each selected transaction.



Transaction Status: success

UUID	8988de839490da24ef75
Connector	1877224IT500525
Method	Creditcard
Type	debit
Created at	2022-01-21 13:11:55
Amount & Currency	20.98 EUR
Fees	
Payout	20.98 EUR
Purchase ID	20220121-8988de839490da24ef75
Merchant TxID	5846a985-bbd7-4e8e-9a60-28b32dcfb65d
Adapter TxID	222522
Success URL	https://moj.a1.si/Blagajna2/Bankart/UpdateTransactionBankart.aspx?transactionId=5846a985-bbd7-4e8e-9a60-28b32dcfb65d
Error URL	https://moj.a1.si/Blagajna2/Bankart/FeedbackFailed.aspx
Cancel URL	https://moj.a1.si/Blagajna2/Bankart/FeedbackFailed.aspx
Postback URL	https://placila.a1.si/api/TransactionNotifications/Bankart

Extra Data

```
{
 "userField4":"000000005914124",
 "userField5":"fkznXDtVBqbYxo7d9E4ZuCmR",
 "xint:cacheHit":false,
 "xint:gwServer":"esh-04",

"3ds:browserAcceptHeader":"text\/html,application\/xhtml+xml,application\/xml;q=0.9,image\/avif,image\/webp,image\/apng,*\/*;q=0.8,application\/signed-exchange;v=b3;q=0.9",
 "3ds:browserIpAddress":"46.123.252.200",
 "3ds:browserJavaEnabled":false,
```

2022-01-21 13:11:55		→	await_redirect
2022-01-21 13:11:55	await_redirect	→	redirected
2022-01-21 13:13:25	redirected	→	await_redirect
2022-01-21 13:13:26	await_redirect	→	redirected
2022-01-21 13:13:54	redirected	→	success

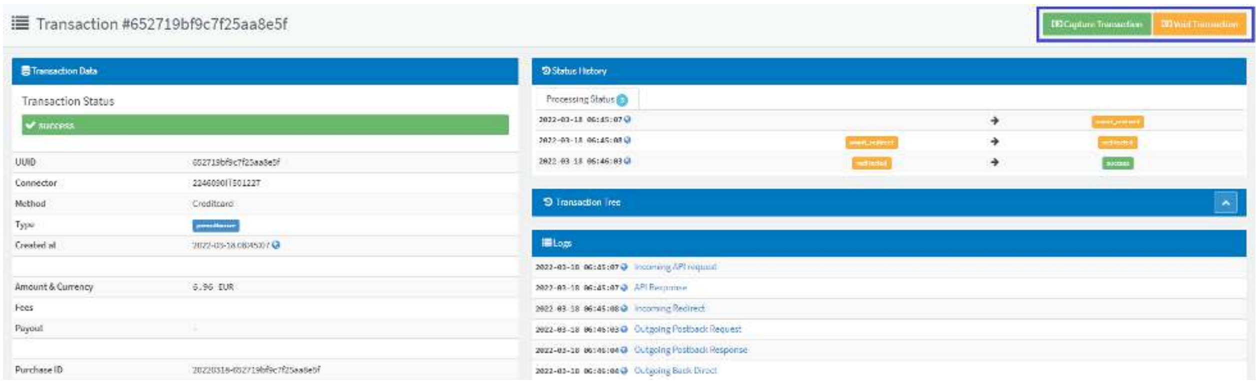
```
2022-01-21 13:11:55 Incoming API request
2022-01-21 13:11:55 API Response
2022-01-21 13:11:55 Incoming Redirect
2022-01-21 13:13:26 Incoming Redirect
2022-01-21 13:13:54 Outgoing Postback Request
2022-01-21 13:13:55 Outgoing Postback Response
2022-01-21 13:13:55 Outgoing Back-Direct
```

Figure 21: Display of page with details on a transaction

Here, in the "Logs" section, developers can check the content of the messages exchanged between the merchant's system and the Payment Gateway system (note: only successfully authenticated messages are visible).

5.2.2. Captures and voids of transactions

In the case of a preauthorization type transaction, you are offered the option of manual capture or void (*highlighted in blue*) within the user interface on the top right.



Transaction Status: success

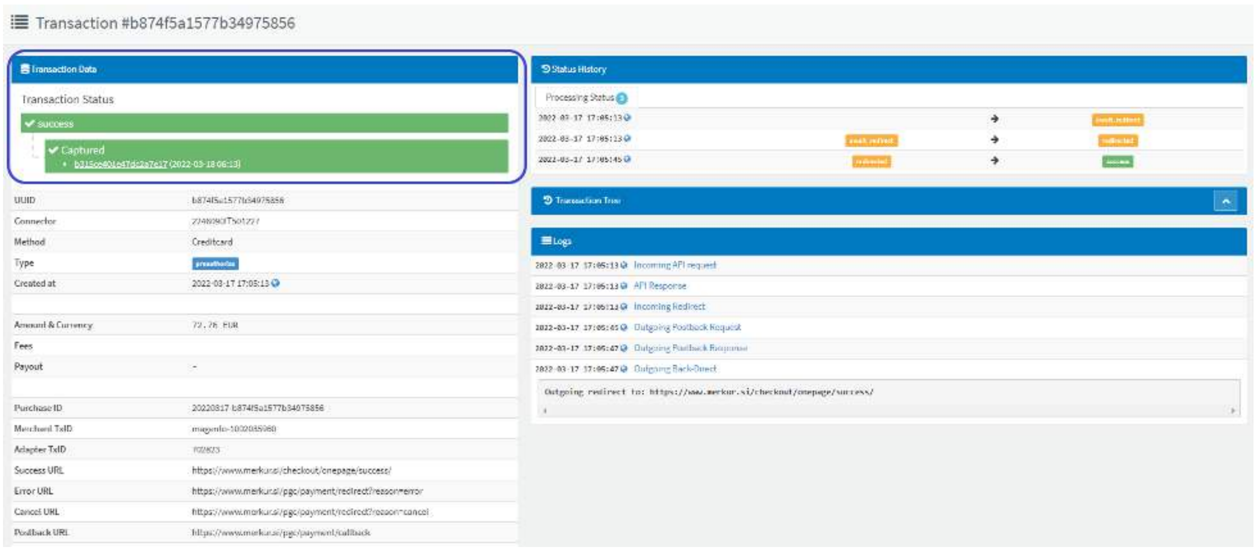
UUID	652719bf9c7f25aa8e5f
Connector	22460901T501227
Method	Creditcard
Type	preauthorize
Created at	2022-03-18 06:45:07
Amount & Currency	6.96 EUR
Fees	
Payout	-
Purchase ID	20220318-652719bf9c7f25aa8e5f

2022-03-18 06:45:07		→	await_redirect
2022-03-18 06:45:08	await_redirect	→	redirected
2022-03-18 06:46:03	redirected	→	success

```
2022-03-18 06:45:07 Incoming API request
2022-03-18 06:45:07 API Response
2022-03-18 06:45:08 Incoming Redirect
2022-03-18 06:46:03 Outgoing Postback Request
2022-03-18 06:46:04 Outgoing Postback Response
2022-03-18 06:46:04 Outgoing Back-Direct
```

Figure 22: Display of the capture/void choice for an individual transaction

Once you have captured the transaction, or the preauthorization has been captured, this will be visible in the "Transaction Status" field. A link to the captured transaction will also be provided. When you click on the link, you will be redirected to that captured transaction.



Transaction #b874f5a1577b34975856

Transaction Data

Transaction Status

success

Captured
- b315ce401e47dc2a7e17 (2022-03-18 06:13)

UUID	b874f5a1577b34975856
Connector	2248090IT501227
Method	Creditcard
Type	preauthorize
Created at	2022-03-17 17:05:13
Amount & Currency	72.76 EUR
Fees	
Payout	-
Purchase ID	20220317-b874f5a1577b34975856
Merchant TxID	magento-1002035060
Adapter TxID	702823
Success URL	https://www.merkur.si/checkout/onepage/success/
Error URL	https://www.merkur.si/pgc/payment/redirect?reason=error
Cancel URL	https://www.merkur.si/pgc/payment/redirect?reason=cancel
Postback URL	https://www.merkur.si/pgc/payment/callback

Status History

Processing Status

2022-03-17 17:05:13		→	await_redirect
2022-03-17 17:05:13	await_redirect	→	redirected
2022-03-17 17:05:45	redirected	→	success

Transaction Tree

Logs

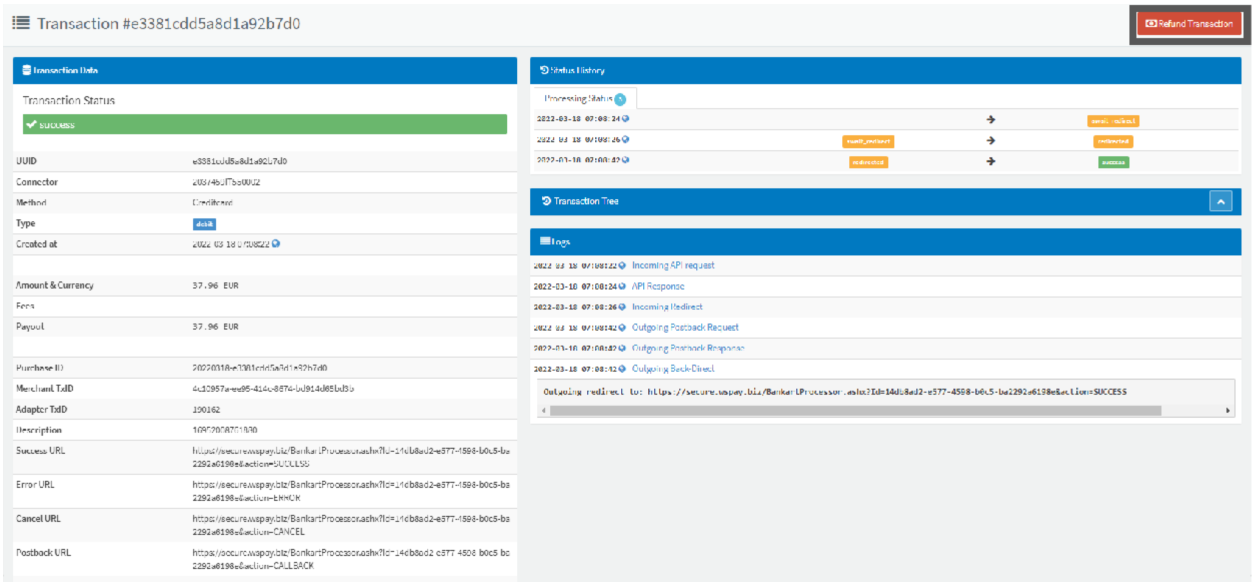
```
2022-03-17 17:05:13 Incoming API request
2022-03-17 17:05:13 API Response
2022-03-17 17:05:13 Incoming Redirect
2022-03-17 17:05:45 Outgoing Postback Request
2022-03-17 17:05:47 Outgoing Postback Response
2022-03-17 17:05:47 Outgoing Back-Direct
Outgoing redirect to: https://www.merkur.si/checkout/onepage/success/
```

Figure 23: Display of the reference (captured, voided) transaction

__Note:__ "fees" and "payout" amounts are not visible. For deposits and commissions, you need to log into the merchant portal, where financial statements are visible.

5.2.3. Refund of the amount of transactions

If the customer is not satisfied with the purchased product or if the product does not reach the customer in a satisfactory condition AND the merchant has the possibility to refund the cost, they can activate this action via the graphical interface upon viewing the transaction details. The "Refund" button will appear on the top right side (*highlighted in dark grey*).



Transaction #e3381cdd5a8d1a92b7d0

Refund Transaction

Transaction Data

Transaction Status

success

UUID	e3381cdd5a8d1a92b7d0
Connector	2037459IT550002
Method	Creditcard
Type	debit
Created at	2022-03-18 07:08:22
Amount & Currency	37.96 EUR
Fees	
Payout	37.96 EUR
Purchase ID	20220318-e3381cdd5a8d1a92b7d0
Merchant TxID	4c10957a-ee95-414c-8874-bd914d65bd3b
Adapter TxID	190162
Description	1095200876198
Success URL	https://secure.wspay.biz/BankartProcessor.ashx?Id=14db8ad2-e577-4598-b0c5-ba2292a6198e&action=SUCCESS
Error URL	https://secure.wspay.biz/BankartProcessor.ashx?Id=14db8ad2-e577-4598-b0c5-ba2292a6198e&action=ERROR
Cancel URL	https://secure.wspay.biz/BankartProcessor.ashx?Id=14db8ad2-e577-4598-b0c5-ba2292a6198e&action=CANCEL
Postback URL	https://secure.wspay.biz/BankartProcessor.ashx?Id=14db8ad2-e577-4598-b0c5-ba2292a6198e&action=CALLBACK

Status History

Processing Status

2022-03-18 07:08:24		→	await_redirect
2022-03-18 07:08:26	await_redirect	→	redirected
2022-03-18 07:08:42	redirected	→	success

Transaction Tree

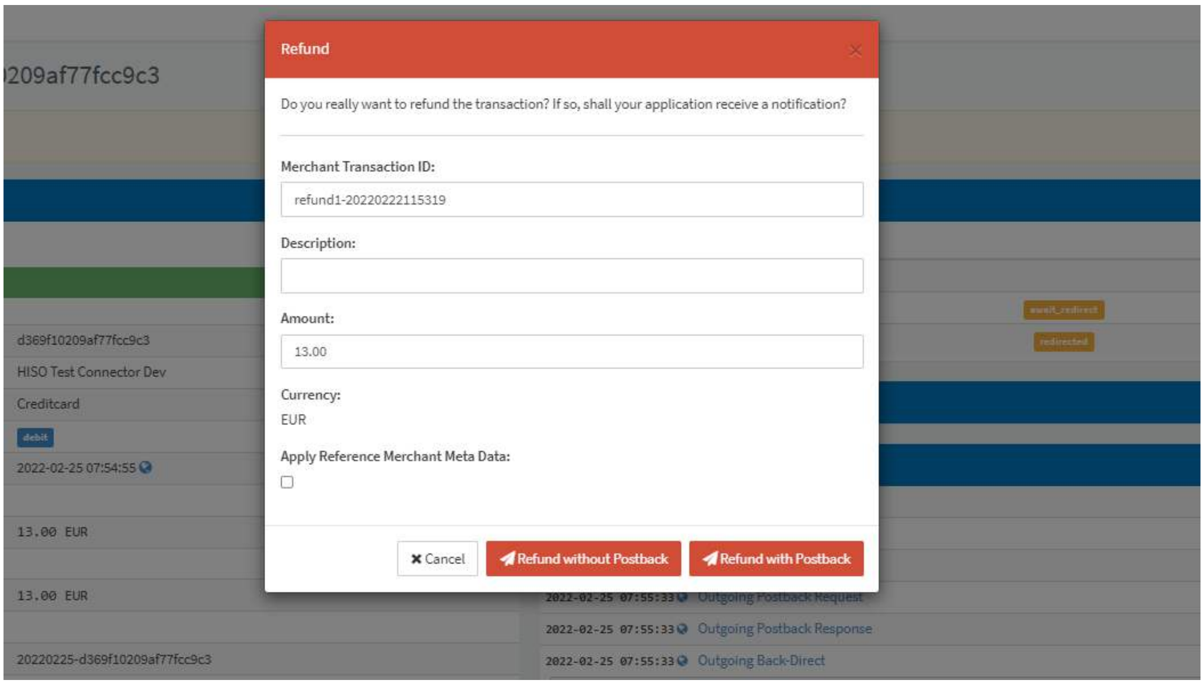
Logs

```
2022-03-18 07:08:22 Incoming API request
2022-03-18 07:08:24 API Response
2022-03-18 07:08:26 Incoming Redirect
2022-03-18 07:08:42 Outgoing Postback Request
2022-03-18 07:08:42 Outgoing Postback Response
2022-03-18 07:08:42 Outgoing Back-Direct
Outgoing redirect to: https://secure.wspay.biz/BankartProcessor.ashx?Id=14db8ad2-e577-4598-b0c5-ba2292a6198e&action=SUCCESS
```

Figure 24: Display of how to execute a refund within the Payment Gateway portal

When you initiate a refund via the user interface, a pop-up will open offering two options: "Refund without Postback" and "Refund with Postback".

The "Refund with Postback" option sends the new transaction status to the merchant's previously defined Callback URL, and if this is not defined, the merchant will NOT receive a change of transaction status when initiating a refund via the Payment Gateway user interface.


Refund

Do you really want to refund the transaction? If so, shall your application receive a notification?

Merchant Transaction ID: refund1-20220222115319

Description:

Amount: 13.00

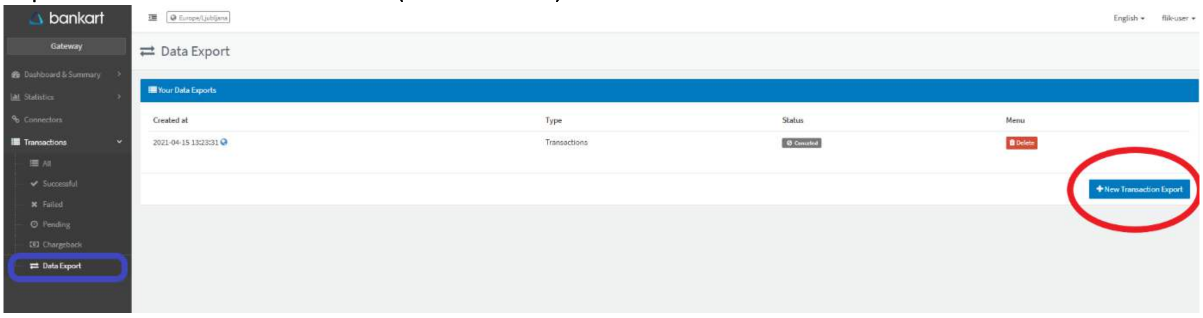
Currency: EUR

Apply Reference Merchant Meta Data: ☐

Figure 25: Screen that appears when a refund is initiated via the portal

5.2.4. Exporting the list of executed transactions

In the **Data Export** subtab (*highlighted in blue in the figure below*) you can filter the transactions and export them to a .cvs or .xlxs file (Excel format).

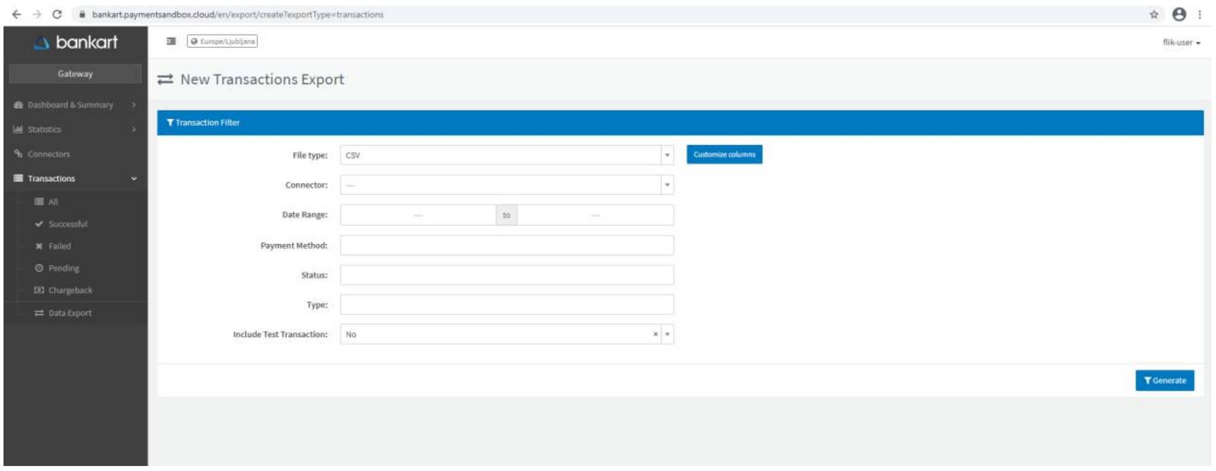

Data Export

Your Data Exports

Created at	Type	Status	Menu
2021-04-15 13:23:31	Transactions	Canceled	Delete

Figure 26: Transaction export screens

Upon clicking the subtab, all documents that have been generated for export up to a certain moment will be displayed. For each export document, the date on which the transactions were exported, the type of data in the export document, the status and the option to Delete or Download are visible.
To export a document, click the "*New Transaction Export*" button (*highlighted in red in the figure above*).



bankart.paymentsandbox.cloud/en/export/create?exportType=transactions

New Transactions Export

Transaction Filter

File type:	CSV
Connector:	—
Date Range:	— to —
Payment Method:	
Status:	
Type:	
Include Test Transaction:	No

Figure 27: Display of screen with filters in transaction export

When the new window (above) opens, select the desired filters and the file type to export and confirm the selection by clicking the "*Generate*" button. You will be redirected to the previous page, where you have to wait a few moments for the document to be generated, and the download/delete option is then offered.
If these options are not displayed after 30 seconds, you can refresh the page yourself.

6. INSTRUCTIONS FOR USING THE PAY BY LINK SERVICE THROUGH THE USER INTERFACE AT BANKART PAYMENT GATEWAY

6.1 Basic info about the Pay by Link service

Pay by Link is a service that is offered as part of Bankart's Payment Gateway and enables the initation of online payments via a link for making the payment. The merchant creates a link for initating the payment in the user interface and sends it to the buyer via the communication channel through which communication takes place between the merchant and the customer (e-mail, SMS message, online conversation or social networks). In the current phase, the created link must be manually copied (by clicking the button) and pasted into the selected communication channel. Automatic sending of link and transaction information via email is still under development.

6.2 Advantages of the Pay by Link service

The advantage of the service is that the merchant does not need technical prior knowledge of the implementation of online payment instruments to use it, as the solution for the merchant is very simple and ready for immediate use. The service is suitable if the merchant does not have his own website or online store and carries out a large proportion of promotions or offer via social networks or phone calls (e.g. influencers, owners of holiday apartments, etc.).

6.3 Flow of a transaction withe the Pay by Link service

The flow of current operation is shown in the diagram below. It is support in the most basic form, and in the following months some functionalities will be added.

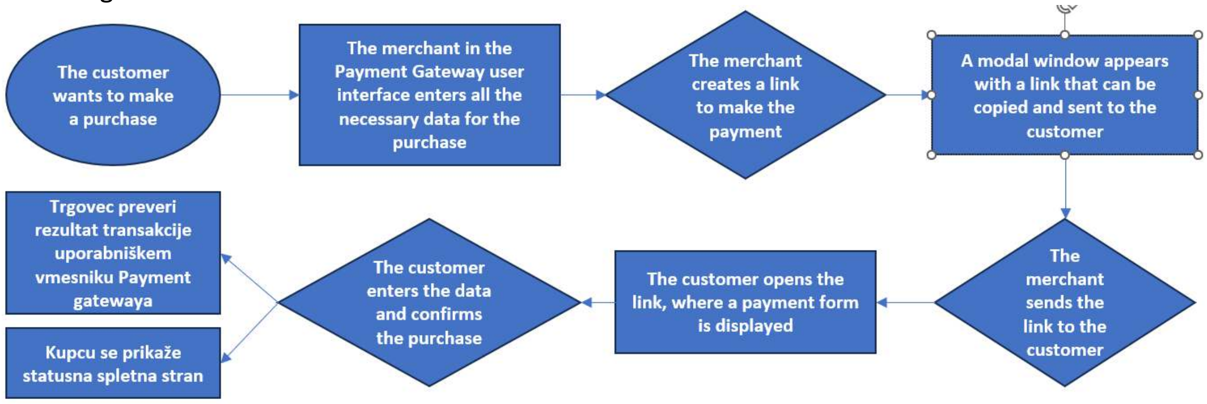
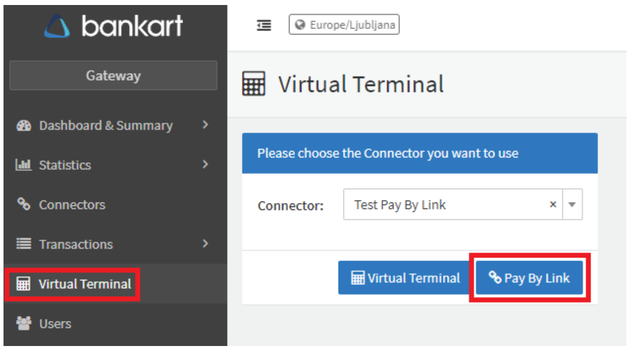

The customer wants to make a purchase
The merchant in the Payment Gateway user interface enters all the necessary data for the purchase
The merchant creates a link to make the payment
A modal window appears with a link that can be copied and sent to the customer
The merchant sends the link to the customer
The customer opens the link, where a payment form is displayed
The customer enters the data and confirms the purchase
Trgovec preveri rezultat transakcije uporabniškem vmesniku Payment gatewaya
Kupcu se prikaže statusna spletna stran

Figure 28: The transaction flow with Pay by Link

6.4 The intiation of the Pay by Link transaction via Bankart's Payment Gateway portal

When the merchant logs into the user interface of Bankart's Payment Gateway (hereinafter: the user interface), a side menu appears on the left side of the screen. In order to use the Pay by Link payment method, the "Virtual Terminal" option is selected, where the section for selecting the connector opens. In the following the connector assigned to use the service is selected and the merchant continues by clicking the "Pay by Link" button (marked in red in the image).

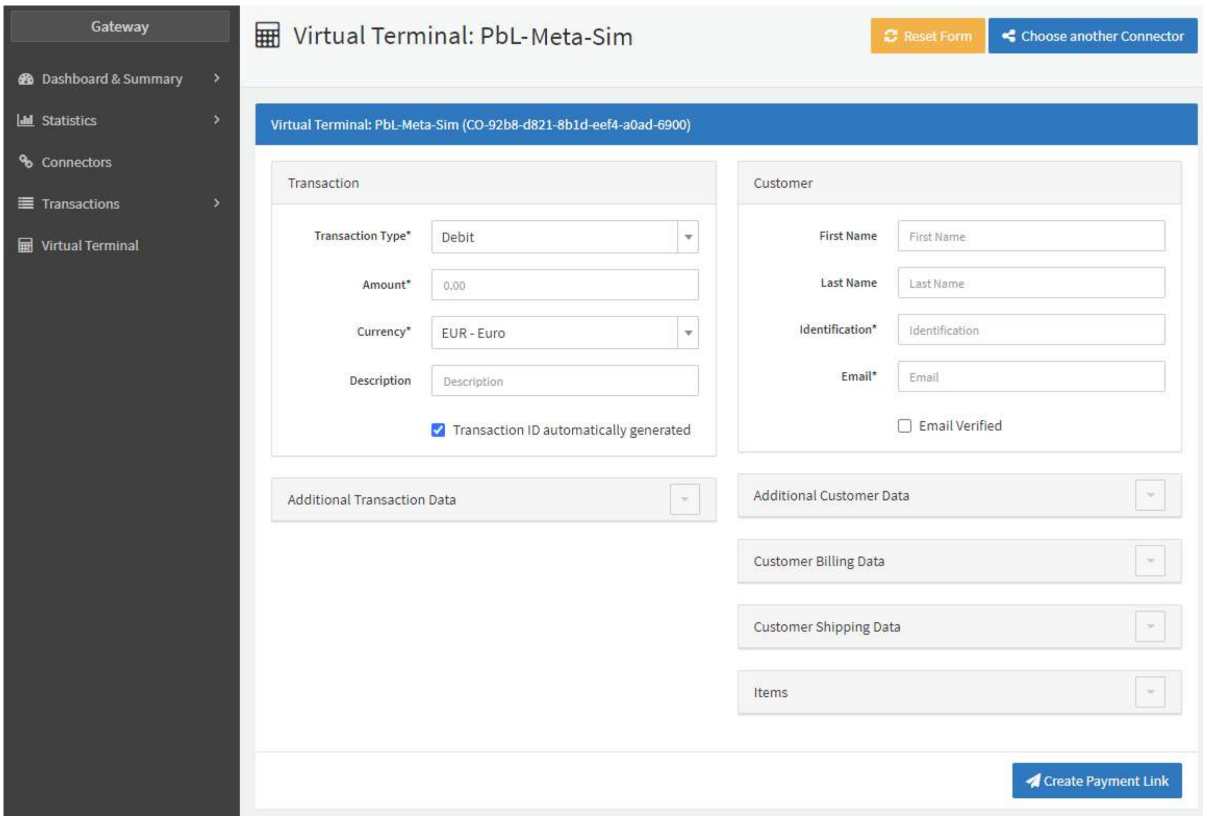


Connector: Test Pay By Link

Figure 29: Where to find the service on the Payment Gateway portal

Sections within the user interface are described in more detail in the next subsection.
Mandatory data for entry by section are:

- **6.5.1 Section** Transaction:
  - Type of transaction supported by the merchant,
  - Amount to be paid,
  - Currency in which the transaction will be triggered (the merchant can only use the currency that the bank allows him to use).
- 6.5.3 Section Customer:
  - Identification of the buyer,
  - Buyer's e-mail address.
- 6.5.4 Section Customer Billing Data:
  - Address 1,
  - City,
  - Post code,
  - Country.



Virtual Terminal: PbL-Meta-Sim (CO-92b8-d821-8b1d-eef4-a0ad-6900)

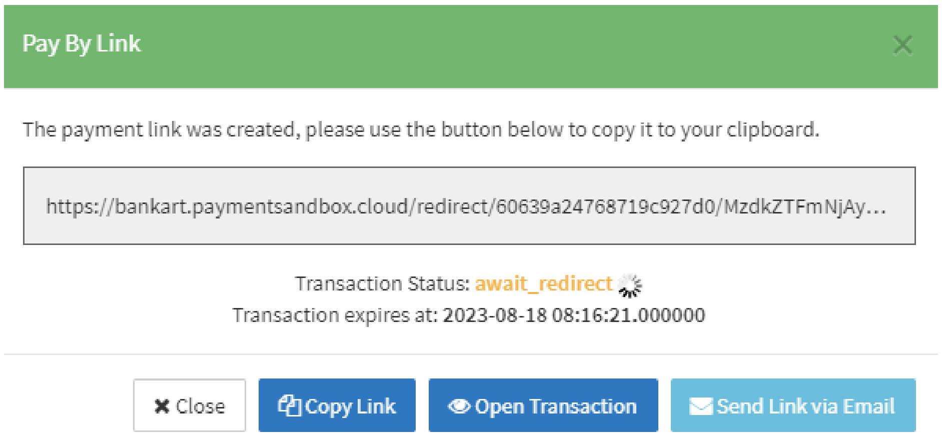
Transaction Type*	Debit
Amount*	0.00
Currency*	EUR - Euro
Description	Description
☑	Transaction ID automatically generated

First Name	First Name
Last Name	Last Name
Identification*	Identification
Email*	Email
☐	Email Verified

Figure 30: Pay by Link console on the Payment Gateway portal

By clicking the "Create Payment Link" button in the lower right corner of the page, a payment link is created and a modal window opens with the following options:

- **Copy Link** – by clicking on the button, the link is copied, which the merchant can then manually paste to the customer in the channel through which the communication takes place,
- **Open Transaction** – by clicking on the the button, the screen with transaction details is opened,
- **Send Link via Email** – by clicking on the button, the link is sent via the email entered in the "Customer" section (currently not available),
- **Close** – by clicking on the button, the modal window closes.


Pay By Link

The payment link was created, please use the button below to copy it to your clipboard.

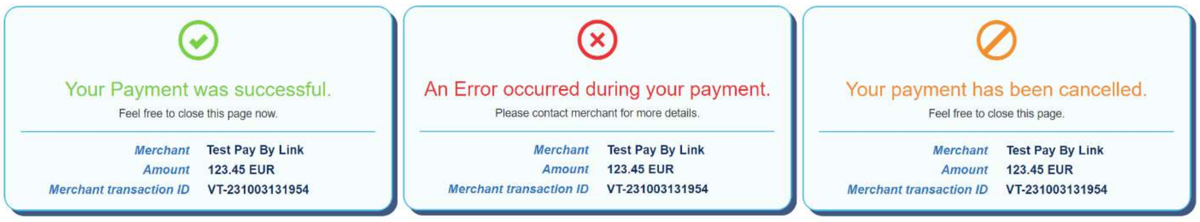
https://bankart.paymentsandbox.cloud/redirect/60639a24768719c927d0/MzdkZTFmNjAy...

Transaction Status: await_redirect
Transaction expires at: 2023-08-18 08:16:21.000000

Figure 31: The popup window with actions you can take

The window also contains information about the status of the transaction and information about the transaction progress. The status of the transaction can always be checked in the tab by clicking on the "Open Transaction" button. In this section additional information about the transaction is also available (what kind of transaction it is, the progress of the connection, whether there are sub-transactions and logs of the transaction itself).
The link in the modal window is copied by the merchant and sent to the customer. By clicking on the link, the customer will open an online payment form, where card details are entered. After entering the data and clicking on the "Submit" button, authorization is carried out. In most cases, Pay by Link transactions require authentication or/and online purchase confirmation. After authentication and authorization, the customer is redirected to the status page, with information about the transaction/payment status:

- The payment was successful,
- The payment was unsuccessful,
- The payment has been cancelled.


Your Payment was successful.
Feel free to close this page now.

Merchant	Test Pay By Link
Amount	123.45 EUR
Merchant transaction ID	VT-231003131954

An Error occurred during your payment.
Please contact merchant for more details.

Merchant	Test Pay By Link
Amount	123.45 EUR
Merchant transaction ID	VT-231003131954

Your payment has been cancelled.
Feel free to close this page.

Merchant	Test Pay By Link
Amount	123.45 EUR
Merchant transaction ID	VT-231003131954

Figure 32: The different result pages at the end of a transaction

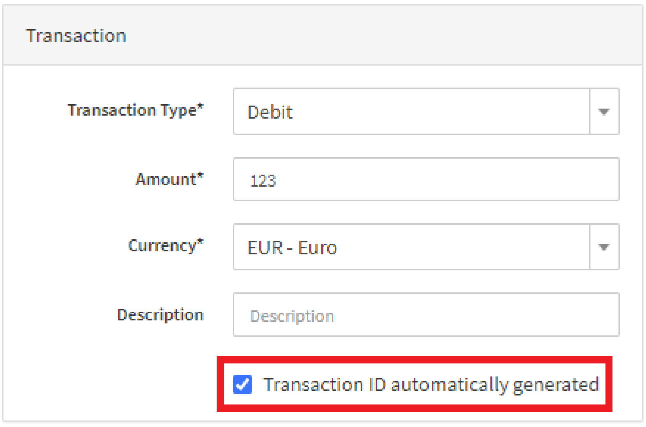
The merchant can check the result of the transaction in the user interface on the page of the executed Pay by Link payment, where there is also an option to review the transaction in detail.

6.5 Detailed description of the data entry sections in the Pay by Link form

All sections relevant to initating a Pay by Link transaction are described here. If the section is not described, then the fields it contains do not need to be filled in, as only these are not relevant for initating the transaction.

6.5.1 Section Transaction

In this section, it is mandatory to fill in the fields already mentioned above. First, the merchant must specify one of the Transaction Type allowed by the bank for the merchant.
It is mandatory to specify the Amount for payment and the Currency of the merchant's country (EUR, MKD, BAM). If a Transaction Type or Currency that is not supported is selected, a warning pop-up window is displayed accordingly. An additional field Description, where data for the merchant's records can be entered as desired, such as the name of the product or service that the merchant is selling.
It is also possible to specify that the transaction ID is automatically generated, if only this is marked with a check mark (marked in red in the picture). If the merchant does not decide for an automatically generated purchase ID, he can define it himself.*Example: The merchant creates an automatically generated transaction on 09/20/2023 at 8:45:18, which the system renames as '«VT-23092084518«. In case the merchant forgot to charge an additional service and wants to charge it additionally later and wishes to have a better record of »related« transactions, he can add »-01« at the end and name the additional transaction »VT-23092084518-01«, which will be named almost the same as the original transaction. The merchant can also define this by one of his own ID keys, which he uses for such an ID.*

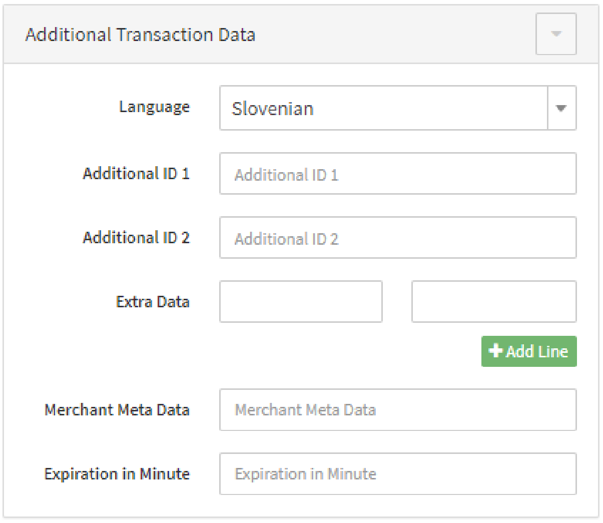

Transaction

Transaction Type*	Debit
Amount*	123
Currency*	EUR - Euro
Description	Description
	☑ Transaction ID automatically generated

Figure 33: The Transaction section by field

6.5.2 Section Additional Transaction Data

There are no mandatory fields in this section, but there are still fields that can affect the transaction. One of such fields is the selection of the language in which the card data input screen is loaded to the customer. This card data input screen is supported in more than 15 different languages.
Other fields are less important and aren't required to be filled for the purpose of initiating a transaction (and are not expected to be filled for now). The exception is the last field "Expiration in Minutes".
If the merchant wants to limit the time of access to the created link, it is important to limit the access of the link to the desired limited number of minutes in the "Expiration in Minutes" field. The merchant enters a duration equal or shorter than the one set in the base: 4320 minutes (or 72 hours/3 days). If this field is empty, by default time the connection will be valid for 72 hours.
Example: if the merchant enters "60" in this field, it will only be valid for 60 minutes and not 4320 minutes (or 72 hours) by default.

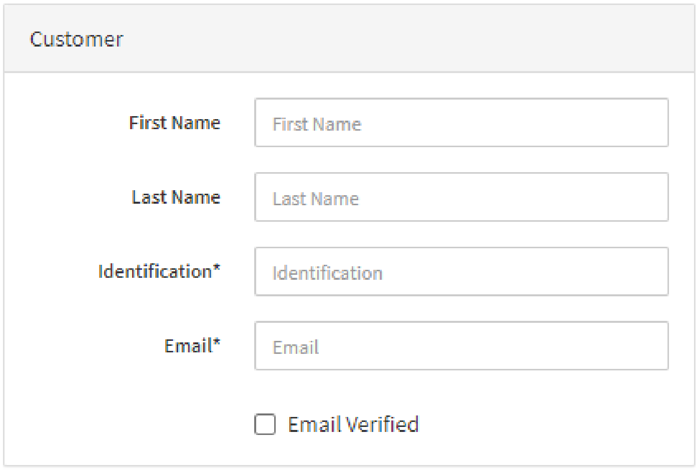

Additional Transaction Data

Language	Slovenian
Additional ID 1	Additional ID 1
Additional ID 2	Additional ID 2
Extra Data	
Merchant Meta Data	Merchant Meta Data
Expiration in Minute	Expiration in Minute

Figure 34: The Additional Transaction Data section by field

6.5.3 Section Customer

In this section, the merchant enters the customer's identification data. First name and last name are not mandatory fields. The mandatory field is the customer's identification (Identification), or in other words, the customer's reference code. With this information, the merchant could more easily identify regular customers and also fill in the fields in the next section "Customer Billing Data". Letters, numbers and symbols can be entered in the field.
The field for entering an Email address is also mandatory. This field is used for automatically sending a link to the transaction, and is also used to process the transaction correctly for card schemes.


Customer

First Name	First Name
Last Name	Last Name
Identification*	Identification
Email*	Email

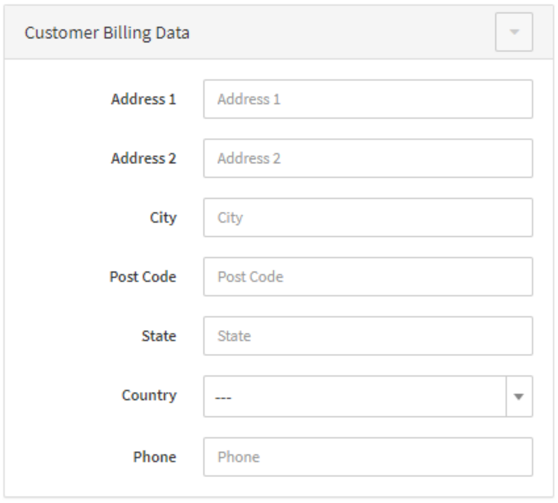
Figure 35: The Customer section by fields

6.5.4 Section Customer Billing Data

Here it is necessary to fill in the fields that determine the customer's place of residence. Within this section, certain fields are mandatory, although not marked with a "*", as this is information that is important for the correct processing of transactions for card schemes and varies between regions.
Mandatory fields that must be entered in Slovenia for successful processing of the transaction are:

- Address 1,
- City,
- Post Code,
- Country.

The merchant can also (optionally) enter the customer's phone number.

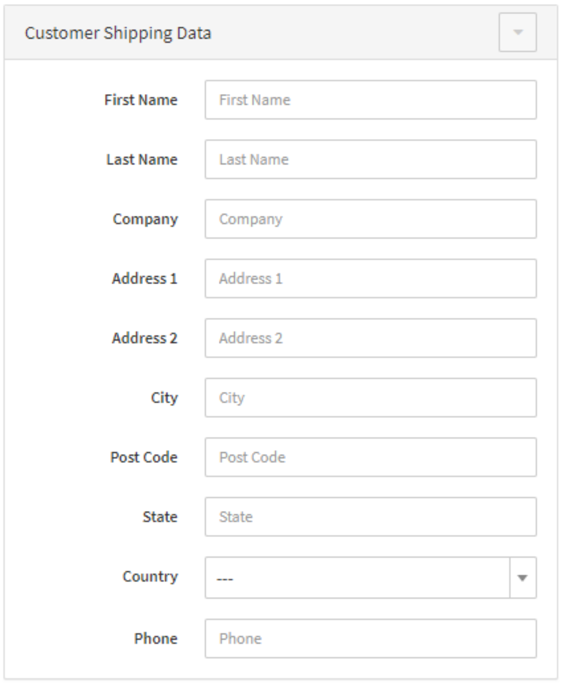


Customer Billing Data	
Address 1	Address 1
Address 2	Address 2
City	City
Post Code	Post Code
State	State
Country	---
Phone	Phone

Figure 36: The Customer Billing Data section by field

6.5.5 Section Customer Shipping Data

All fields in this section are optional. These are filled in if the delivery is addressed to another person or to a company other than the information provided in the " Customer Billing Data" section.

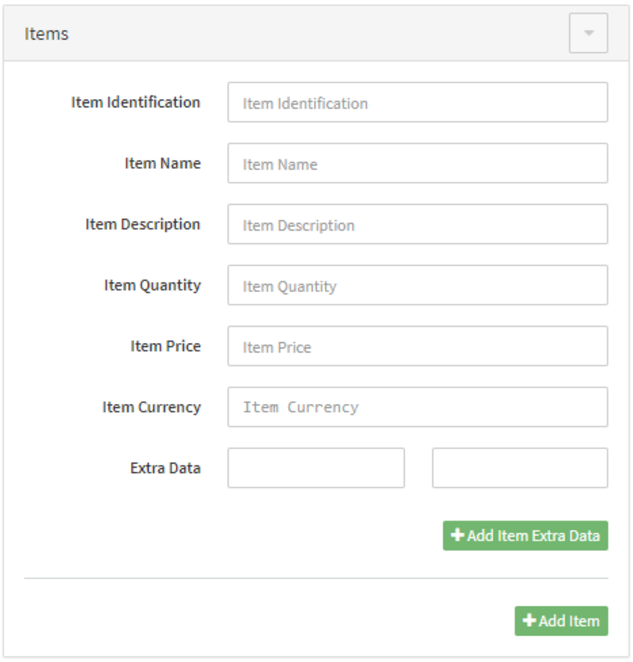


Customer Shipping Data	
First Name	First Name
Last Name	Last Name
Company	Company
Address 1	Address 1
Address 2	Address 2
City	City
Post Code	Post Code
State	State
Country	---
Phone	Phone

Figure 37: The Customer Shipping Data section by field

6.5.6 Section Items

Also in this section, all fields are optional. In this section, it is possible to add some basic data about the products/services sold. You can enter the number of the identification item,name of the item, description of the item, quantity of the item for each individual purchased product, price of the item and also the currency of the item. By clicking the "Add Item" button, it is possible to add several additional items.
The data entered in this section will not be displayed anywhere at the moment to the customer and will only be displayed to the merchant within the user interface on the transaction overview page Later, when it is supported, it will be displayed in an automatically generated Email when the payment connection is created.



Items

Item Identification	Item Identification
Item Name	Item Name
Item Description	Item Description
Item Quantity	Item Quantity
Item Price	Item Price
Item Currency	Item Currency
Extra Data	

Figure 38: The Items section by field

7. INSTRUCTIONS FOR USING THE PAY BY LINK SERVICE THROUGH AN API CALL

For the successful implementation of the Pay by Link API service (referred to as PBL API) on the merchant's side, basic knowledge of using the Payment Gateway interface, the Pay by Link service, and an understanding of API functionality is required. This document provides a detailed, step-by-step guide on how to use the PBL API service.

7.1 Mandatory and optional parameters in the PBL API call

When initiating a PBL transaction via an API call, it is basically a debit transaction, meaning a POST call is used to trigger the debit transaction (as described in the Payment Gateway documentation above). The key difference in the API call is the inclusion of an additional array named *payByLink*, which contains new parameters (marked in blue in the table). This distinguishes it from a standard debit call.

Ime polja	Opis	Obvezen podatek
amount	Amount of transaction	YES
currency	Currency used for transaction	YES
successUrl	Page displayed to the customer after a successfully completed transaction	YES
cancelUrl	Page displayed to the customer when the transaction is canceled by the customer	YES
errorUrl	Page displayed to the customer when the transaction fails for any reason	YES
callbackUrl	URL where the merchant receives the transaction result and data	YES
sendByEmail	By default, the value is set to "false". More details can be found below in the subsection.	YES
merchantTransactionId	ID of the transaction on the merchant's side.	YES
billingAddress1	Obtaining card details from the customer (described on page 8, click the underlined text)	YES
billingCity	Obtaining card details from the customer (described on page 8, click the underlined text)	YES
billingPostcode	Obtaining card details from the customer (described on page 8, click the underlined text)	YES
billingCountry	Obtaining card details from the customer (described on page 8, click the underlined text)	YES
email	Email Adress of the cardholder	YES
firstName	Name of the cardholder	NO
lastName	Last name of the cardholder	NO
language	The language in which the Hosted Payment Page is displayed	NO
expirationInMinute	Transaction validity limit. For more information, see the subsection below.	NO

Example of API call in JSON format:

```
{
            "merchantTransactionId": "20250425123456",
            "amount": "9.99",
            "currency": "EUR",
            "successUrl": "https://example.si/en/finalize/SUCCESS=1",
            "cancelUrl": " https://example.si/en/finalize/CANCEL=1",
            "errorUrl": " https://example.si/en/finalize/ERROR=1",
            "callbackUrl": " https://example.si/en/finalize/postback",
            "customer": {
                      "firstName": "Janez",
                      "lastName": "Novak",
                      "billingAddress1": "Test address 1",
                      "billingCity": "Ljubljana",
                      "billingPostcode": "1000",
                      "billingCountry": "SI",
                      "email": "janez.novak@customer.si",
                      "ipAddress": "98.765.432.10"
},
            "language": "en",
            "payByLink": {
                    "sendByEmail": false,
                     "expirationInMinute": 120
            }
}
```

7.2 Additional Parameters in the PBL API call (payByLink array) - transaction initiation

In the API call, an additional array *payByLink* is sent, containing two extra fields, which are described in detail below.

7.2.1 Field sendByEmail

If a merchant wants to trigger Pay by Link transactions through and API call, they must include the following (mandatory fields) in the API request **"sendByEmail": false** and **"email":" janez.novak@customer.si".** This means the merchant will send the email containing the transaction link from their own domain/address. The link to the transaction (Hosted Payment Page) is provided in the **"redirectUrl"** field, from which the merchant can copy the link and include it in the email.
Example of "*redirectUrl*": https://gateway.bankart.si/3ds/challenge/1234a567bc89def0ghij).

7.2.2 expirationInMinute

If the merchant includes the (optional field) **"expirationInMinute": 60**, the transaction validity is limited to **60 minutes**. If the provided value exceeds Bankart's default limit of 4320 minutes (72 hours or 3 days), the transaction will be restricted to 4320 minutes.
Additional documentation with a detailed description of API fields is available at the following link: https://gateway.bankart.si/documentation/apiv3?php#transaction-data-pay-by-link.

7.3 Additional parameters in the PBL API (payByLinkData array) – transaction response

The data in the *payByLinkData* array is important because it informs the merchant about the transaction's expiration time and provides the URL that must be called if they wish to cancel the transaction before it expires.

Example of an API JSON response highlighting the mentioned array:

```
"payByLinkData ": {
        "expiresAt": "2025-04-03 12:34:56 UTC",
        "cancelUrl": "cancelUrl": " https://example.si/en/finalize/CANCEL=1"
}
```

7.3.1 Field expiresAt

If the merchant includes the "expirationInMinute" value in the API request, the API response will contain the "expiresAt" parameter, indicating the exact time until which the transaction link remains active.

7.3.2 Field cancelUrl

The merchant also has the option to cancel an initiated Pay by Link transaction by making an API call to the "cancelUrl", which is provided in the API response (the same URL the merchant initially sent).

Important Notice: For technical reasons, canceling a Pay by Link transaction does not always prevent its completion of the transaction. If the customer has already opened the Hosted Payment Page, they can still proceed with finalizing the payment.

A transaction can only be canceled before the customer opens the link and initiates the process of payment. To avoid this, we recommend using the "expirationInMinute" parameter to appropriately limit the transaction validity based on your needs.

7.4 Sending the link to the customer

Sending the link from the bank's or Bankart's server is currently not supported. However, you can still copy and paste the link without any technical restrictions and share it with the customer through any communication channel of your choice.
You can send the link to the customer via:

- Email,
- Social media,
- The merchant's website,
- Or any other communication channel.

When the customer clicks on the link, they will be redirected to a secure Hosted Payment Page (HPP), where they can enter their card details.

7.5 Status of successful, failed, and canceled transactions

The status of successful, failed, and canceled transactions is mandatory in the API request. Merchants can select the language in which the Hosted Payment Page and transaction status page will be displayed to the customer. If the merchant does not specify a language or provides an unsupported language code, the Hosted Payment Page will default to English. The currently supported languages are listed on page 8, section 3.4.1. Designing a payment page with an input mask (above).

Types of URL addresses and their meanings:

- **Successful Transaction**: If the customer completes the payment successfully, they will be redirected to **success_url**. On this page, the status page confirming a successful payment must be displayed (page 35, image 32).
- **Failed Transaction**: If a technical error occurs during the transaction, the customer will be redirected to **errorUrl**. On this page, the status page indicating a failed payment must be displayed (refer to (page 35, image 32).
- **Canceled Transaction**: If the customer cancels the payment, they will be redirected to **cancel_url**. On this page, the status page confirming the cancellation must be displayed (page 35, image 32).

7.6 Monitoring payment status via Callback URL

To monitor the transaction status, you need to set up a **callback_url**, where you will receive notifications about the transaction status (e.g., successful payment, failed payment, etc.). For more details, refer to page 7, section 3.2. Technical documentation and requirements (above).