K&H Payment Services Ltd

vPOS / Payment Gateway

Integration document

v1.2

Table of Contents

Purpose of the system

The system enables online bankcard payments via the web shops of e-merchants contracted with K&H Payment Services Ltd.

characteristics of the virtual POS service:

- forint, euro and US dollar (HUF, EUR, USD) based settlement
- user-friendly payment page available in multiple languages
- highly secure transactions executed on the encrypted payment page of K&H Payment Services Ltd
- bank card frauds are reduced to the minimum
  the system runs an automatic blacklist check on the bank card in question and it also checks whether it is valid and if there are sufficient funds for the intended transaction
- highly secure standardised solutions
  the various components of the system communicate with each other using an authentication-based encryption method with a high level of protection based on the standardised PKI infrastructure (strong encryption, SSL, digital signature)
- clean architecture, secure operation
  the simple interfaces, the condition-based operation and the internal structure of the system guarantee the secure execution of transactions and continuous operation
- additional services
  in addition to payments, refund transactions are also available

Integrating your web shop with K&H Payment Gateway

1. K&H PS assigns a web shop MID and a vPOS ID to your web shop and sends them to you. Please use the provided web shop MID in the URLs in both environments (sandbox/production environment).
2. Create the protocol provided in the description in the sandbox (sandbox) of your web shop application, using professional assistance if required.
3. Send the return URLS for the sandbox to vpos_khpos@kh.hu.
   Scenario 1: when the result of a successful transaction is displayed
   Scenario 2 (optional): when a transaction has failed or been cancelled
4. Generate a key pair in the application provided by K&H PS. The public key will be automatically assigned to your vPOS ID.
5. Test the application, including the digital signature, then send the information required for testing to K&H PS. At the same time please send us the information we need in order to verify whether your web shop has the required contents.
6. K&H PS completes the functional testing of bank card transactions and verifies if your web shop has the required contents.
7. If everything is found in order, the payment gateway will be enabled in the production environment, of which you will be notified by email.
8. Generate a key pair for the production environment using the application provided by K&H PS. Once it is activated you can access the production payment gateway of K&H PS by overwriting the URL.

Requirements for the web shop

- ability to handle the following scenarios: payment failed, payment cancelled, communication cut off, card holder does not return to web shop
- ability to handle refund transactions
- ability to issue confirmations by email to card holders about all successful transactions (payment, refund), containing the following details:
  - transaction ID (txid)
  - amount (amount)
  - currency (ccy)
  - bank authorisation number – in the fourth line of the result card received on calling the PGResult page
  - full name of merchant
  - web address of merchant
  - description of goods/services

Required website content

https://khpos.hu/sw/static/file/eloirt_honlap_tartalom.pdf

Payment process

1. When a card holder gets to the point in the process where they must pay for the selected goods or services in your web shop, they click on the Pay button, which triggers your web shop server to submit a code via the URL on the Pay button (pl. CGI script, ASP or servlet). This code, which forms part of the web shop application, generates a 302 type HTTP response redirecting to **URL1**, with the following parameters:
   - unique transaction ID (txid – maximum 10 numeric characters), defined by the merchant and preventing multiple payments for the same transaction;
   - transaction type (sale);
   - web shop MID (mid = 12345678),
   - amount payable (in the case of HUF: in fillér, rounded to the next whole forint amount);
   - currency code;
   - signature;
   - language code.

The web shop MID is a code given by K&H PS following contract signature. The digital signature is generated by the web shop application using the key generated by the merchant, which protects the transaction ID, the transaction type, the web shop MID, the amount payable and the currency code.

2. Following redirection the browser calls **URL1** using the parameters received, similarly to the example below (which can be interpret in the sandbox):

https://pay.sandbox.khpos.hu/pay/v1/PGPayment?txid=3141592653&type=PU&mid=10234506&amount=1234000&ccy=HUF&sign=a1154ffeb7…535cfc88cfd784&lang=HU

The servlet initiated by **URL1** verifies the uniqueness of the requested transaction and the authenticity of the signature based on the transaction ID received as a parameter. If everything is found in order, a response is generated in the language determined by the language code, which contains the parameters of the payment transaction (amount, currency code, full name of merchant) and prompts for a card number, expiry date and CVV2. The card holder then clicks on the Pay button to start the requested transaction.

3. Once the transaction has been processed, the payment page redirects the card holder to the specified return URL, to which the system adds a txid = parameter.

4. Irrespective of this, the web shop can query the result of the transaction by calling URL2. Call syntax:

https://pay.sandbox.khpos.hu/pay/v1/PGResult?mid=10234506&txid=3141592653

Transactions must be queried using TLS 1.2 or higher protocol.

The result is in plain text format, with fixed structure records, which contain the transaction status code (line 1) and, following authorisation, the authorisation response code (line 2), the text message for the response code (line 3) and the bank authorisation number (line 4).

```
ACK
0
ELFOGADVA / ENGEDELYEZVE
08304J
```

Possible transaction codes:

- "NAK" - payment failed (e.g. due to insufficient funds on the account)
- "UTX" - transaction ID unknown
- "PEN" - payment pending, call again
- "ERR" - error (e.g. signature not authentic)
- "CAN" - card holder clicked on the Cancel button
- "EXP" - time for payment expired (after 25-30 minutes)
- "ACK" - payment successful

Refund

1. You may decide to refund the full amount of a successful payment transaction, or a part thereof, to the card holder. **Refund transactions may be initiated from the day after the completion of the original payment transaction.** In this case the web shop calls **URL1** with the following parameters:

- transaction ID of the transaction to be refunded, defined by the merchant;
- transaction type (refund) **type=RE**;
- merchant ID;
- amount to be refunded (in filler, rounded to forint) [may be less than the amount of the original transaction];
- currency code,
- signature.

The digital signature is generated by the web shop application using the key generated by the merchant, which protects the transaction ID, the transaction type, the web shop MID, the amount payable and the currency code.

https://pay.sandbox.khpos.hu/pay/v1/PGPayment?txid=3141592653&type=RE&mid=10234506&amount=1234000&ccy=HUF&sign=a1154ffeb7…535cfc88cfd784

2. Based on the transaction ID received as a parameter, the servlet invoked by **URL1** checks whether the transaction in question exists, if it is in “ACK” status, the amount specified, the currency code and the authenticity of the signature. If everything is in order, then the Payment Gateway generates an result card containing the parameters in question (transaction ID, amount, currency code, web shop MID).

3. To query the result of the refund transaction, call **URL2**. Syntax:

https://pay.sandbox.khpos.hu/pay/v1/PGResult?mid=10234506&txid=3141592653

The result card is in plain text format. Possible contents:

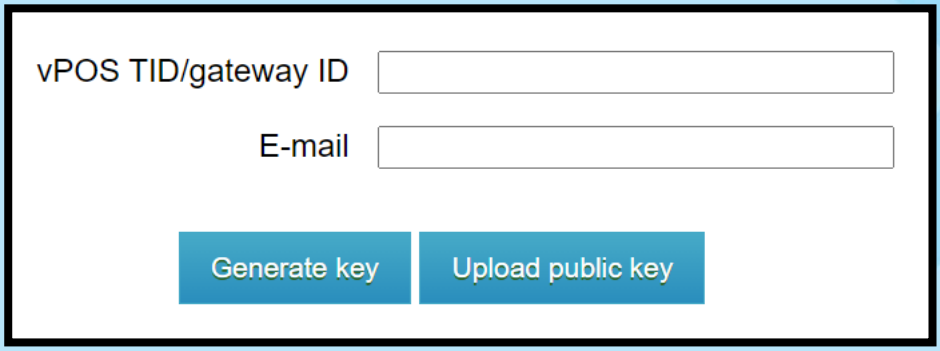
- "UTX" - unknown transaction ID
- "PE2" - refund pending, repeat query
- "ERR" error (e.g. the refund was initiated on the transaction date)
- "VOI" refunded

Description of the web shop interface

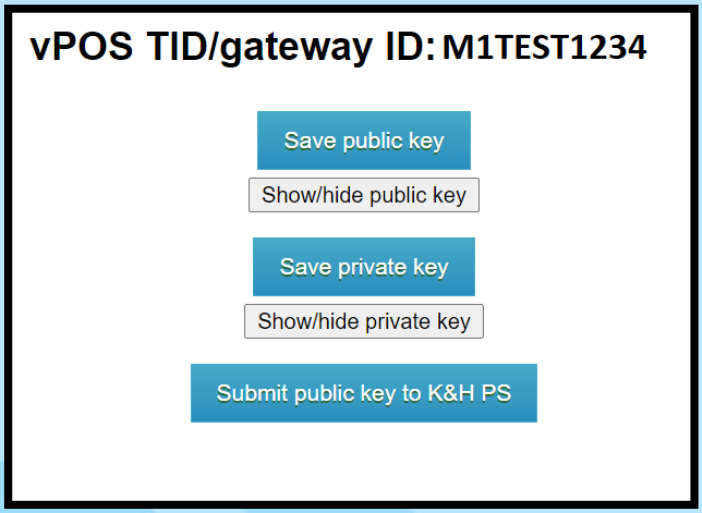
a. generating a key

Different key pairs must be used in the sandbox and in the production environment. They can be generated using a simple online application, which can be found here:

Sandbox: https://sandbox.khpos.hu/keygen
Production environment: https://pay.khpos.hu/keygen



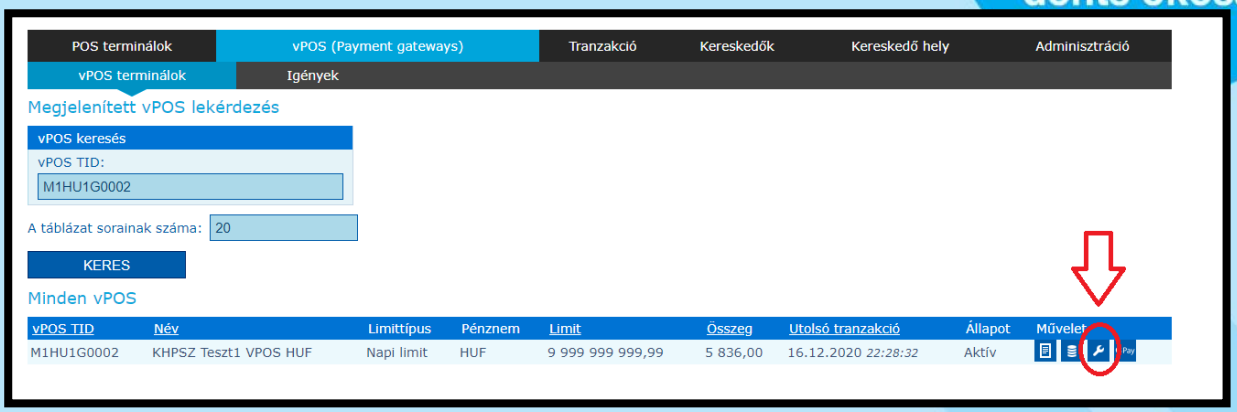
- enter the vPOS TID/gateway ID and the **technical email address specified in your agreement**
- generate key

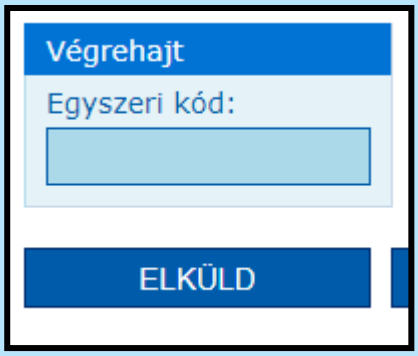


- save public key, save private key
- submit public key to K&H PS

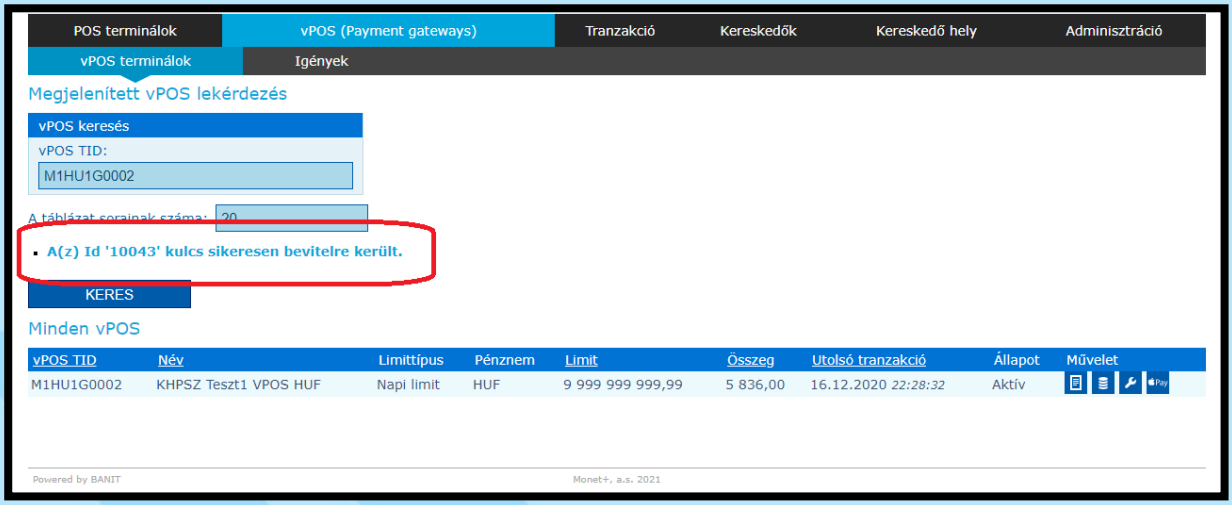
The key submitted to the sandbox can be used straight away.

The production environment sends an automatic message to the technical email address provided in response to the key submitted, which contains a one-off activation code.
Activate the generated key pair in the POS24 application using the code received.
(vPOS (Payment gateways) / vPOS terminals)





If the activation is successful, the following message will be displayed, after which the key can be used in the production environment straight away:



b. generating a signature:

```
$ java -classpath <lásd lent> RSASign sign "mid=1&txid=100&type=PU&amount=1&ccy=HUF" test_private_key "test_password"
```

`data_to_sign` : **data to sign: parameters of the PGPayment call `in` GET style (without a language parameter) in this order: `mid, txid, type, amount, ccy`. For example: `mid=1&txid=100&type=PU&amount=1000&ccy=HUF`). The parameters in the signature must be in the above order.**

`test_private_key`: the file containing the private key.
The output is the signature itself in hexadump format, as the `sign` parameter must be specified as a URL parameter.

OpenSSL can also be used for the signature:

openssl dgst -sign private_key.pem -hex -sha1 <file_containing_the_stuff_to_sign >sign.hex

In this example the file_containing_the_stuff_to_sign file contains the text to be signed as per the above, without line and file end characters.

If the services of the openSSL can also be accessed from a script language as well as from a command line, then of course a signature can also come from there, for example:

PHP:

```
#!/usr/bin/php
<?php

$data = "mid=1&txid=100&type=PU&amount=1000&ccy=HUF";

$fp = fopen("./private_key.pem", "r");
$priv_key = fread($fp, 8192);
fclose($fp);
$pkeyid = openssl_get_privatekey($priv_key);

// compute signature
openssl_sign($data, $signature, $pkeyid);

// free the key from memory
openssl_free_key($pkeyid);

echo bin2hex($signature);
?>
```

Remarks:

The programs work with SUN Java virtual machine 1.4 or later. The separator in the `classpath` parameter depends on the platform, i.e.
in Windows:

```
bcprov-jdk15-146.jar;khb_sign_util.jar
```

in Unix (including Linux):

```
bcprov-jdk15-146.jar:khb_sign_util.jar
```

`bcprov-*.jar` is the correct version of the Bouncy Castle cryptography package. Please ensure that you choose the correct package for the JDK version in the event of an upgrade.

http://www.bouncycastle.org/

http://www.bouncycastle.org/latest_releases.html

Examples with Java 1.5 and 1.6:

Java 1.5:

```
java15 -classpath bcprov-jdk15-146.jar;khb_sign_util.jar RSASign
keygen test_private_key "test_password" test_public_key
```

Java 1.6:

```
java16 -classpath bcprov-jdk16-146.jar;khb_sign_util.jar RSASign
keygen test_private_key "test_password" test_public_key
```

where java15 and java16 stand for the Java machine in question, but first install:
Java(TM) Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files
You can find it here:

http://java.sun.com/javase/downloads/index.jsp
http://www.oracle.com/technetwork/java/javase/downloads/index.html

If you did not install it, you will receive this error message: `java.lang.SecurityException`

Access to `bcprov-jdk*.jars`:
`http://www.bouncycastle.org/latest_releases.html`

All signature functions can also be accessed directly from Java calling RSASign tool class methods:

```
import java.security.PrivateKey;
import java.security.interfaces.RSAPublicKey;
public static void writeKey(
            PrivateKey key,
            String private_key_file_name,
            String password ) throws Exception;

public static void generateKey(
            String private_key_file_name,
```

```
            String public_key_file_name,
            String password ) throws Exception;

public static PrivateKey readKey(
            String private_key_file_name,
            String password ) throws Exception;

public static void writePublicKey(
            RSAPublicKey key,
            PrivateKey priKey,
            String public_key_file_name ) throws Exception;

public static RSAPublicKey readPublicKey(
            String public_key_file_name ) throws Exception;

public static final byte[] sign(
            PrivateKey key,
            byte data[] ) throws Exception;

public static final boolean verify(
            PublicKey key,
            byte data[],
                      byte sigBytes[] ) throws Exception;
```

URLs to be called by the shop application

The web shop can call the following URLs (also using the GET or POST process of the HTTP protocol).
We recommend that you use the POST process for security reasons.

The URLs of the Payment Gateway can be found here:
https://pay.sandbox.khpos.hu/pay/v1 for the sandbox and
https://pay.khpos.hu/pay/v1 for the production system.

For payment transactions:

URL	function	mandatory parameters	optional parameters	response
/PGPayment (**URL1**)	start new transaction	mid, txid, type=PU, amount, ccy, sign	lang	Text/html type response for user

/PGResult (**URL2**)	query transaction result: https://pay.sandbox.khpos.hu/pay/v1/PGResult?mid=12345678&txid=1234567890	mid, txid		Text/plain type response with a fixed length structure:
				PEN - payment pending, call again
				CAN - card holder clicked on the Cancel button
				EXP - time for payment expired (after 25-30 minutes)
				NAK - payment failed (e.g. due to insufficient funds on the account).
				UTX - transaction ID unknown
				ERR - error
				ACK - payment successful
				`ACK`
				`0`
				`ELFOGADVA / ENGEDELYEZVE`
				`08304J`
				3 char status
				3 num response code
				48 char plain text message
				8 char authorisation number

For refund transactions:

URL	function	mandatory parameters	optional parameters	response
/PGPayment (**URL1**)	refund transaction	mid, txid, type=RE, amount, ccy, sign		Text/plain type response with a fixed length structure: **PE2** – request accepted. **UTX** – unknown transaction ID. **ERR** - error (e.g. incorrect signature, transaction details do not match).
/PGResult (**URL2**)	query transaction status	mid, txid		Text/plain type response with a fixed length structure: **PE2** – refund pending. **UTX** – transaction ID unknown. **ERR** - error **VOI** - refunded. 3 char status, 3 numeric response code, 48 char plain text message, 6 char authorisation number (change from the original).

Data dictionary: parameters and their explanation

Name	Type	
Transaction ID (txid)	(maximum) 10 numeric char	Provided by merchant; must not start with 0!
Transaction Type (type)	(maximum) 2 char	Provided by merchant PU - sale RE – refund
Merchant ID (mid)	(maximum) 10 num	Provided by the Bank
Transaction amount in fillér (amount)	num (00)	Maximum value: 4294967200 must end with ‘00’
Currency code (ccy)	3 char	HUF, EUR, USD
Signature (sign)	256 char	Signature generated by merchant
Language (lang)	2 char	HU, EN, DE, ES, IT, PL etc.

other

K&H PS stores the following parameters about the web shop:

- merchant ID (mid);
- name, address and contact details of merchant;
- return URL (sandbox/production) provided by the merchant in the case of successful transactions (ACK);
- return URL (sandbox/production) provided by the merchant in the case of unsuccessful transactions.

The stability of the system is guaranteed by the following rules:

- K&H PG only accepts a request for a new transaction if the **txid_mid** is unique. This ensures that payment transactions pending cannot revert in the status graph for any reason (for example if the 'Back' button is pressed in the browser)
- a transaction may be aborted in any status, it will not have any consequences for the operation of K&H PG
- having submitted a request for a new payment transaction, the merchant can subsequently query its current status at any time using the **txid**
- the system accepts refund requests only for transactions in "ACK" status

Language codes

“lang” is an optional parameter, whose purpose is to specify the language of the response by the system.

The value of the parameter is a two-letter language code conforming to the ISO 639-1 standard.

List of supported language codes:

HU	Hungarian
DE	German
ES	Spanish
EN	English
FR	French
IT	Italian
PL	Polish
PT	Portuguese
RO	Romanian
SK	Slovakian

Users can choose from additional languages on the payment page.

(Hungarian, Croatian, Czech, English, French, German, Italian, Japanese, Polish, Portuguese, Romanian, Russian, Slovakian, Slovenian, Spanish, Turkish, Vietnamese)

Testing

To be checked – general

- The Bank's payment page must not appear in an inline frame (iframe) in the web shop application or in a popup window.
- The amount displayed on the Bank's payment page must be correct.
- The transaction ID must not be longer than 10 characters.

The payment page shows the name, address and contact details of the web shop. Please check if this information is correct.

Once you have successfully integrated your web shop application with K&H PG, please run the following tests:

Payment successful (status: ACK)

Once you have been directed to the payment page, enter the following card details:
Card number: 4154610001000209
Expiry date: 10/23
CVC: 100
Please wait until you are automatically redirected to your own web shop.

Expected functioning: the web shop must display a message to the effect that the payment transaction was successful and send a confirmation by email.
Items to be checked:
- amount payable;
- the result of the transaction is displayed in the web shop;
- the customer has received the e-mail;
- the contents of the e-mail (transaction ID [txid], amount (amount), currency (ccy), bank authorisation number, full name of merchant (acquirer), web address of merchant (acquirer), description of goods/services).

Transaction rejected (status: NAK)

Once you have been directed to the payment page, enter the following card details:
Card number: 5542860001000224
Expiry: 06/23
CVC: 200
Once the payment has been rejected click on the return button to be redirected to your own web shop. As a result of the "NAK" message, the transaction can be queried after the 25-30 minutes allowed for pending.

Expected functioning: the web shop must display a message to the effect that the payment failed.
Items to be checked:
- the result of the transaction is displayed in the web shop.

Return to web shop without payment (status: CAN)

Do not enter any card details, just click on the return button to be redirected to your own web shop.

Expected functioning: the web shop must display a message to the effect that the payment failed.
Items to be checked:
- the result of the transaction is displayed in the web shop.

Refund (status: VOI)

Initiate a refund following a successful payment.
Refund transactions **may be** initiated **from the day after the completion of the original payment transaction.**

Expected functioning: if the case of a successful refund (VOI) the web shop must send a confirmation to the card holder by email.
Items to be checked:
- the customer has received the e-mail;
- the contents of the e-mail (transaction ID [txid], amount (amount), currency (ccy), [modified] bank authorisation number for the refund, full name of merchant (acquirer), web address of merchant (acquirer), description of goods/services).

Contact information, notifications:

If you have successfully run the above tests in your sandbox please let us know in an email message sent to vpos_khpos@kh.hu. Please put your vPOS **web shop MID** and the **web address of the contracted web shop** in the subject field.
If the return URLs are different in the sandbox and your production environment, please also include the production return URLs in your message.

Please note that during testing we will check if your web shop has the required contents (Required web shop content) to please make sure that all the necessary information is available!

Once we receive your notification we will inform you about the go-live schedule.

Access to the production system:

Once the production system has been given authorisation, remove the sandbox part from the beginning of the URLs used in the sandbox.
The test card details provided for testing will not be valid in the production environment.

Example:

https://pay.sandbox.khpos.hu/pay/v1 access to sandbox
https://pay.khpos.hu/pay/v1 access to the production environment