MERCHANT INTEGRATION

3D PAY HOSTING MODEL

Version 1.4

08 February 2024

Version	Date	Description
1.3	25 June 2012	Added hidden encoding parameter.
1.4	08 February 2024	Hash calculation method is updated

Contents

3D Pay Hosting Model

3D Pay Hosting model is the basic internet integration model with payment page hosting, supporting 3D transactions.

Basic Properties:

- Enables processing of 3D secure card transactions
- HTTP Post method for merchant integration – Payment is done automatically by Nestpay.

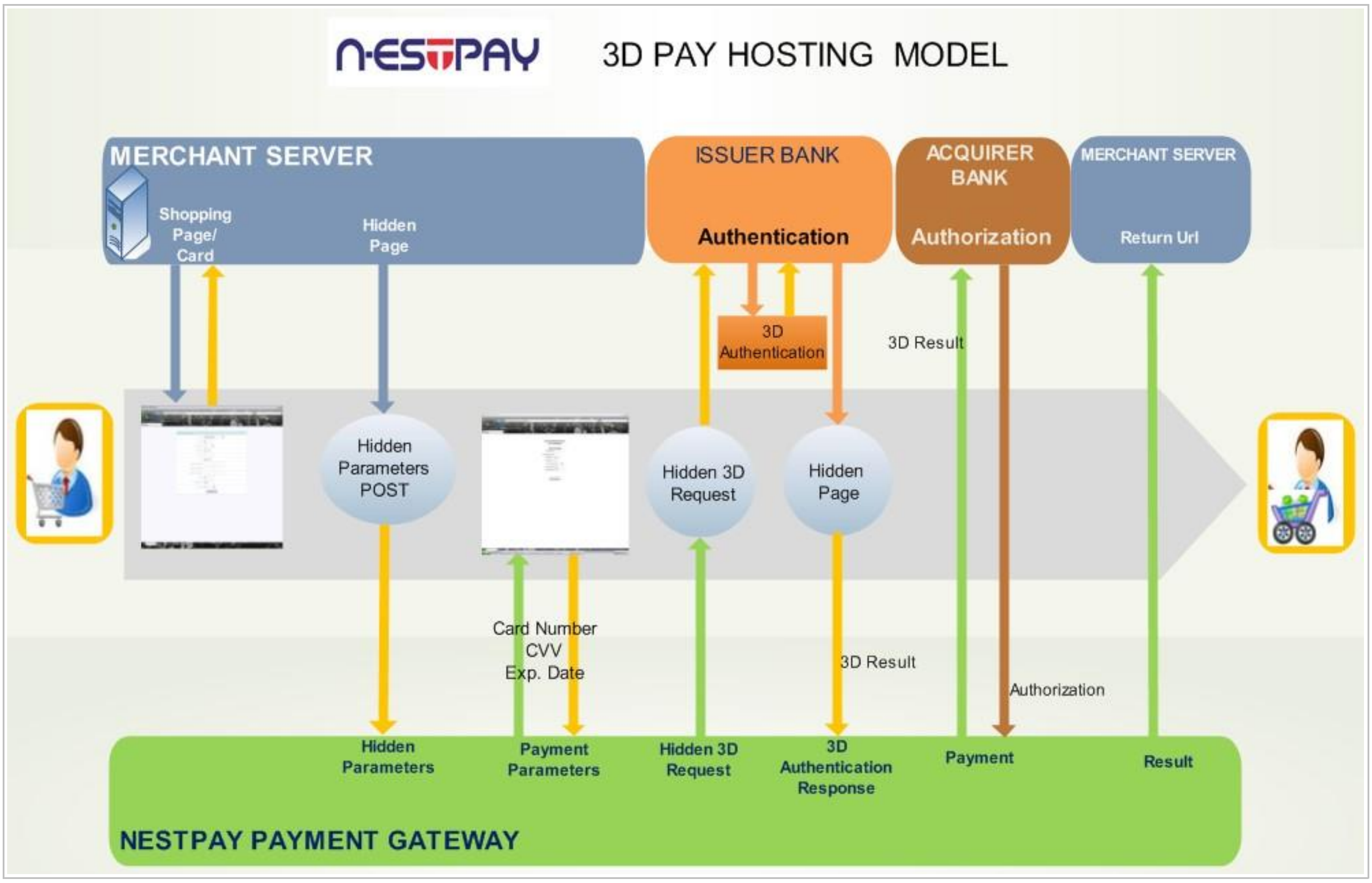
After obtaining all necessary shopping data from customer (like order amount, currency, customer name/surname etc.), merchant server generates a unique order ID.

Necessary parameters are posted with HTTP Post method to Nestpay gateway.

For card payment methods (Visa, MasterCard etc.) merchant server needs to submit the card details like card number, CVV2, and expiry date information. After the order/card data is obtained from the user the 3D flow (enrollment and authentication queries) starts. In 3D flow, the 3D authentication information of the customer is queried by the issuer bank. The methods for 3D authentication can be different for different issuers. Examples of 3D authentication methods are usage of 3D secure password, one-time password, security questions.

1. The customer knows that his/her personal information is not saved by the merchant, because credit card information is queried by Nestpay, not the merchant.
2. Integration process is easy.
3. Bank's SSL certificate is used. Therefore the software cannot be updated.
4. In addition to the obligatory parameters, merchant can POST its own data, such as username, user email or user id. Those data is sent back to the merchant by the bank.

Nestpay 3D Pay Hosting Model



PayHosting Model Diagram

Quick Start Guide

Making successful sale VISA transaction with **3D PayHosting** Model.

Generate Hash for Client Authentication

After the merchant receives the parameters, merchant shall check the parameters on the merchant server to verify them. The Hash must be validated. To ensure hash control, the message is sent only from Payten.

As of 2023, Payten has switched to hashing with SHA-512 base encoding.

Creating Plain Text for Hash

“|” in the data created for Hash Version 3, the character serves as a separator between parameters. While creating the data to be hashed for Hash Version 3, all parameters sent to Payten are used for hash calculation. While these parameters are included in the hash calculation, the parameter names are listed alphabetically from A to Z and separated by "|" By adding a separator, the data to be hashed in the relevant alphabetical order is created. While preparing the data to be hashed, even if a parameter is sent to Payten as empty, it is added to the relevant data (For the empty value, see the use of the Installation parameter when calculating the hash in the example below).

Then, add “|” to the end of the data prepared alphabetically. The Merchant Secure Key (storeKey) is added using the separator.

Important Note: If "|" character is used inside the value of the parameters, this character is also used to separate parameters. In order not to be confused with the character “|” in the value of the parameter character “\|” when creating hash data is changed to . In addition, if there is a "\" character in the value of the parameters, the relevant "\" characters must be replaced with the value "\\" to avoid confusion. **For instance,**

Original Value	: ORDER-256712jbs\j6b|
Value introduced to Hash	: ORDER-256712jbs\\j6b\\|

Sample Parameters and Hash Calculations:

clientId	100200127 "
amount	95.93
okurl	http://localhost:8080/SampleCodeJSPTest/GenericVer3ResponseHandler
failUrl	http://localhost:8080/SampleCodeJSPTest/GenericVer3ResponseHandler
TranType	Auth
Instalment	
callbackUrl	http://localhost:8080/SampleCodeJSPTest/GateResponseControl.jsp
currency	949
rnd	87954458746
storeType	3D_PAY_HOSTING

lang	tr
hashAlgorithm	ver3
BillToName	name
BillTocompany	billToCompany
refreshTime	5
storeKey	TEST1234

□ Hash

Order of Use of Parameters Used in Hash Data:

amount|BillToCompany|BillToName|callbackUrl|clientid|currency|failUrl|hashAlgorithm|Instalment|lang|okurl|refreshtime|rnd|storetype|TranType|storeKey

Plaintext:

95.93|billToCompany|name|http://localhost:8080/SampleCodeJSPTest/GateResponseControl.jsp|100200127|949|http://localhost:8080/SampleCodeJSPTest/GenericVer3ResponseHandler|ver3||tr|http://localhost:8080/SampleCodeJSPTest/GenericVer3ResponseHandler|5|87954458746|3D|Auth|TEST1234

Hash = Base64(SHA512(plaintext))

Important Note II: Parameters named **“encoding” & “hash”** will not be taken into account in calculating the hash.

All values that replace these parameters are added in the same order. The resulting hashed text is encoded with the base64 version according to the SHA512 algorithm. Under normal circumstances, the hash text produced must be the same as the HASH parameter value published by Payten. Otherwise, the merchant must contact the Payten support team.

Example: non-3D card transactions

Assuming the action has response parameters:

clientid, oid, AuthCode, ProcReturnCode, Response, rnd

HASHALGORITHM: ver03

HASH: CVJssbkrhIzqZXVTwGobciDZI+A=

The resulting hash must be the same as the hash value in the return of the HASH parameter.

Posting hidden Parameters

Posting the mandatory input parameters to Nestpay Payment Gateway located at **https://host/fim/est3dgate** as hidden parameters.

clientid **:** Merchant ID (given by Nestpay) **storetype**
: “3d_pay_hosting”
hash **:** Hash value for client authentication **islemtipi** **:** "Auth" **amount**
: amount transaction amount **currency** **:** ISO code of transaction currency (949 for TL) **oid** **:** Unique identifier of the order **encoding** **:** UTF-8 **okUrl**
: The return URL to which **Nestpay Payment Gateway** redirects the browser of the customer if transaction is completed successfully.
failUrl **:** The return URL to which **Nestpay Payment Gateway** redirects the browser of the customer if transaction is completed unsuccessfully.
lang **:** Language of the payment pages hosted by Nestpay ("tr" for Turkish, "en" for English)
pan **:** Card number
Ecom_Payment_Card_ExpDate_Year **:** Expiry year
Ecom_Payment_Card_ExpDate_Month **:** Expiry month

Sample HTTP form with mandatory parameter set

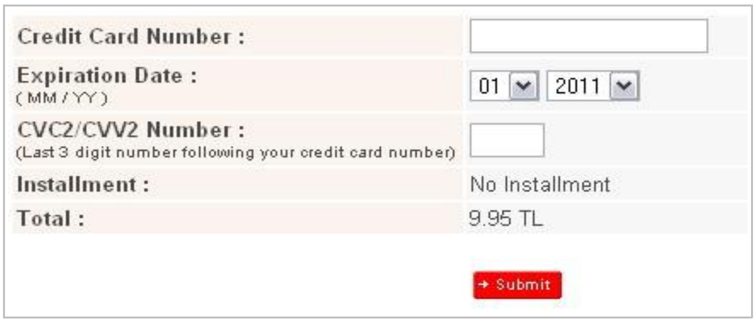
Please post the following parameters to Nestpay Gateway as a hidden parameter with HTTP form.

`<input type="hidden" name="encoding" value="UTF-8">` mandatory parameter solves the encoding problem of the payment and return pages during the transaction action.

```
<form method="post" action="https://host/fim/est3dgate">
       <input type="hidden" name="clientid" value="990000000000001"/>
       <input type="hidden" name="storetype" value="3d_pay_hosting" />
       <input type="hidden" name="hash" value="iej6cPOjDd4IKqXWQEznXWqLzLI=" />
       <input type="hidden" name="islemtipi" value="Auth" />
       <input type="hidden" name="amount" value="91.96" />
       <input type="hidden" name="currency" value="949" />
       <input type="hidden" name="oid" value="1291899411421" />
        <input type="hidden" name="encoding" value="UTF-8">
        <input type="hidden" name="okUrl" value="https://www.teststore.com/success.php"/>
       <input type="hidden" name="failUrl" value="https://www.teststore.com/fail.php" />
       <input type="hidden" name="lang" value="en" />
       <input type="hidden" name="rnd" value="asdf" />
       <input type="hidden" input name="pan" value="4242424242424242">
       <input type="hidden" input name="Ecom_Payment_Card_ExpDate_Year" value="28" >
       <input type="hidden" input name="Ecom_Payment_Card_ExpDate_Month" value="10">
</form>
```

VISA Payment Page

Consumer will enter his/her card details to complete the transaction and clicks the Pay button.



Credit Card Number :	
Expiration Date : (MM / YY)	01 2011
CVC2/CVV2 Number : (Last 3 digit number following your credit card number)	
Installment :	No Installment
Total :	9.95 TL

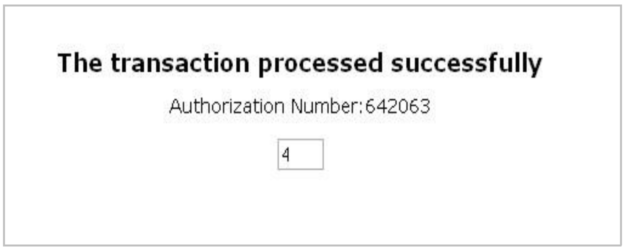
Fig-1

3D Authentication

In 3D flow, the 3D authentication information of the customer is queried by the issuer bank. The methods for 3D authentication can be different for different issuers. Examples of 3D authentication methods are usage of 3D secure password, one-time password, security questions.

Transaction Result Page

The transaction result will be displayed to customer. If the transaction is successful the authorization code will be displayed. The customer will be redirected to okUrl if refreshtime is over.



The transaction processed successfully

Authorization Number:642063

4

Fig-2

Merchant Success Page

If the transaction is successful the customer will be redirected to **okUrl**, which is submitted on step 2 to Nestpay Payment Gateway. All parameters posted by merchant returns back the merchant. In addition to merchant parameters, gateway returns the transaction response parameters and MPI response parameters (related to 3D secure transaction flow) which can be found in Appendix A.

Basic transaction response parameters for full authenticated successful 3D transaction:

Response	: "**Approved**"
AuthCode	: Authorization code of the transaction
HostRefNum	: Host reference number
ProcReturnCode	: "00"
TransId	: Unique transaction ID
mdStatus	: "1"

For the example transaction above the transaction response parameters would be:

Response	: "**Approved**"
AuthCode	: 544889
HostRefNum	: 034910000320
ProcReturnCode	: "00"
TransId	: 103491153310910033
mdStatus	: "1"

Integration Basics

HTTP Post Integration

After receiving a valid order parameters are post to Nestpay payment gateway as hidden parameters with HTTP form. In addition to mandatory parameters merchant can post order billing/shipping and order item details to payment gateway which can be viewed later on Merchant Administration Panel. For optional parameters explanations please refer to Appendix – A.

The 28 byte-long base-64 encoded xid parameter is the unique Internet transaction ID which is required for 3D secure transactions. If it is not sent by the merchant, it will be created automatically by the system.

Sample HTTP form with mandatory and optional parameters

```
<form method="post" action="https://host/fim/Nestpaygate">

       <input type="hidden" name="clientid" value="990000000000001"/>
       <input type="hidden" name="storetype" value="3d_pay_hosting" />
       <input type="hidden" name="hash" value="iej6cPOjDd4IKqXWQEznXWqLzLI=" />
       <input type="hidden" name="islemtipi" value="Auth" />
       <input type="hidden" name="amount" value="91.96" />
       <input type="hidden" name="currency" value="949" />
       <input type="hidden" name="oid" value="1291899411421" />
        <input type="hidden" name="encoding" value="UTF-8">
       <input type="hidden" name="okUrl" value="https://www.teststore.com/success.php"
       />
       <input type="hidden" name="failUrl" value="https://www.teststore.com/fail.php" />
       <input type="hidden" name="lang" value="tr" />
       <input type="hidden" name="rnd" value="asdf" />
       <input type="hidden" input name="pan" value="4242424242424242">
       <input type="hidden" input name="Ecom_Payment_Card_ExpDate_Year" value="28" >
       <input type="hidden" input name="Ecom_Payment_Card_ExpDate_Month" value="10">
       <input type="hidden" name="xid" value="egsF658v9uNpdqmksFZ5j9xHV/U=" />

</form>
```

<!-- Billing Parameters [All Optional]-->

```
 <input type="hidden" name="tel" value="012345678">
 <input type="hidden" name="Email" value="test@test.com">
 <input type="hidden" name="firmaadi" value="Billing Company">
 <input type="hidden" name="Faturafirma" value="John Smith">
 <input type="hidden" name="Fadres" value="Address line 1">
 <input type="hidden" name="Fadres2" value="Address line 2">
 <input type="hidden" name="Filce" value="Warsaw">
 <input type="hidden" name="Fil" value="mystate">
 <input type="hidden" name="Fpostakodu" value="12345">
<input type="hidden" name="Fulkekodu" value="400">
```

<!-- Shipping Parameters [All Optional]-->

```
<input type="hidden" name="NakliyeFirma" value="Shipping Company">
<input type="hidden" name="tismi" value="John Smith">
```

```
<input type="hidden" name="tadres" value="Address line 1">
<input type="hidden" name="tadres2" value="Address line 2">
<input type="hidden" name="tilce" value="Warsaw">
<input type="hidden" name="til" value="mystate">
<input type="hidden" name="tpostakodu" value="12345">
<input type="hidden" name="tulkekod" value="400">
```

<!-- Order Item Parameters [All Optional]-->

```
 <input type="hidden" name="ItemNumber1"
e="a5"> input type="hidden" name="ProductCode1"
e="a5"> input type="hidden" name="Qty1" value="3">
t type="hidden" name="Desc1" value="a5 desc"> input
="hidden" name="Id1" value="a5"> input type="hidden"
="Price1" value="6.25"> input type="hidden"
="Total1" value="7.50">
```

```
<
<
<
<
<
<
```

Card Transactions

Submitting the form with card data will start the 3D authentication flow with the customer. After the 3D authentication process is completed the MPI response parameters and all parameters sent by merchant will be post back to merchant to make the payment. The payment will be done according to **mdStatus** field which is shows the status code of the 3D secure transaction.

MPI Response Parameters

mdStatus : Status code for the 3D transaction
txstatus : 3D status for archival
eci : Electronic Commerce Indicator
cavv : Cardholder Authentication Verification Value, determined by ACS.
md : Hash replacing card number
mdErrorMsg : Error Message from MPI

Possible *mdStatus* Values

- 1 = Authenticated transaction (Full 3D)
- 2, 3, 4 = Card not participating or attempt (Half 3D)
- 5, 6, 7, 8 = Authentication not available or system error
- 0 = Authentication failed

Successful Transaction

The authorization code will be displayed. The customer will be redirected to **okUrl** of merchant server if refreshtime is over. All input parameters along with transaction response parameters will be post to **okUrl**, the Response parameter will be "**Approved**"

Failed Transaction

The failure message will be displayed. The customer will be redirected to **failUrl** of merchant server if refreshtime is over. All input parameters along with transaction response parameters will be post to **failUrl**, the Response parameter will be "**Declined**" or "**Error**".

Transaction Response Parameters

Response : "Approved", "Declined" or "Error"
AuthCode : Authorization code of the transaction
HostRefNum : Host reference number
ProcReturnCode : Transaction status code
TransId : Unique transaction ID
ErrMsg : Error text (if Response "Declined" or "Error")
ClientIp : IP address of the customer
ReturnOid : Returned order ID, must be same as input oid
MaskedPan : Masked credit card number
PaymentMethod : Payment method of the transaction
rnd : Random string, will be used for hash comparison
HASHALGORITHMS : Hash algorithm, used to calculate the hash.
HASH : Hash value of merchant password field and etc.

MPI Response Parameters

mdStatus : Status code for the 3D transaction
txstatus : 3D status for archival
eci : Electronic Commerce Indicator
cavv : Cardholder Authentication Verification Value, determined by ACS.
mdErrorMsg : Error Message from MPI (if any)
xid : Unique Internet transaction ID

Possible Transaction Results

- **Response:** "Approved"

  ProcReturnCode will be "00". This shows that the transaction has been authorized.

- **Response:** "Declined"

  ProcReturnCode will be a 2 digit number other then "00" and "99" which corresponds to acquirer error code. This shows that the transaction has NOT been authorized by the acquirer. ErrMsg parameter will give the detailed description of the error. For detail description of acquirer error codes for *ProcReturnCode* refer to Appendix B.

- **Response:** "Error"

  ProcReturnCode will be "99". This shows that the transaction has NOT reached to acquirer authorization step. ErrMsg parameter will give the detailed description of the error.

Hash Checking

Once the merchant receives the parameters, a Hash must be checked on the merchant server to verify the parameters. To ensure hash control, the message is sent only from Payten.

Generating the plain text for hash:

The parameters used in hash calculation are as follows:

amount|BillToCompany|BillToName|callbackUrl|clientid|currency|failUrl|hashAlgorithm|Instalment|lang|okurl|refreshtime|rnd|storetype|TranType|storeKey

Depending on the type of transaction, a subset of the following parameters will be included as hash generation:

• Non-3D card transactions:

amount|BillToCompany|BillToName|callbackUrl|clientid|currency|failUrl|hashAlgorithm|Instalment|lang|okurl|refreshtime|storetype|TranType|storeKey

• 3D secure card transactions:

amount|BillToCompany|BillToName|callbackUrl|clientid|currency|failUrl|hashAlgorithm|Instalment|lang|okurl|refreshtime|rnd|storetype|TranType|storeKey

All values that replace these parameters are added in the same order. The merchant password is appended as a final value to the end of this string. The resulting hashed text is encoded with base encoding version 64 according to the SHA512 algorithm. Under normal circumstances, the hash text produced must be the same as the HASH parameter value published by Payten. Otherwise, the merchant must contact the Payten support team.

Example: non-3D card transactions

Assuming the action has response parameters:

HASHALGORITHM: ver3

HASH: CVJssbkrhIzqZXVTwGobciDZI+A=

Code Samples

The following procedure for 3D Pay Hosting Model areas. Values test purposes had been inserted. 3D Pay Hosting Model on edited code examples. Merchants, taking into account variables must define values for them. These codes reference purpose formed.

ASP Code Sample

.Net Code Sample

JSP Code Sample

PHP Code Sample

APPENDIX A: Gateway Parameters

Mandatory Input Parameters

Parameter	Description	Format
clientid	Merchant ID	Maximum 15 characters
storetype	Merchant payment model	Possible values: "pay_hosting", “3d_pay”, "3d", "3d_pay_hosting"
islemtipi	Transaction type	Set to "Auth" for authorization, “PreAuth” for preauthorization
amount	amount transaction amount	Use "." or "," as decimal separator, do not use grouping character
currency		ISO code of transaction currency 3 characters (example: 949 for TL)
oid	Unique identifier of the order	Maximum 64 characters
encoding	encoding parameter	UTF-8
pan	Card number	Maximum 20 digits
Ecom_Payment_Ca	Card expiry year	4 digits rd_ExpDate_Year
Ecom_Payment_Ca	Card expiry month	2 digits rd_ExpDate_Month

okUrl	The return URL to which Nestpay redirects the customer if transaction is completed successfully.	Example: http://www.test.com/ok.php
failUrl	The return URL to which Nestpay redirects the customer if transaction is completed unsuccessfully.	Example: http://www.test.com/fail.php
lang	Language of the payment pages hosted by Nestpay	"tr" for Turkish, "en" for English
rnd	Random string, will be used for hash comparison	Fixed length, 20 characters
Hashalgorithm	Hash algorithm value, used for calculation	Should be 'Ver3.'
hash	Hash value for client authentication	

Optional Input Parameters

Parameter	Description	Format
refreshtime	Redirection counter value(to okUrl or failUrl) in seconds.	
description	description	Maximum 255 characters
taksit	Instalment count	Number
xid	Unique internet transaction ID	28 characters, base64 encoded
Email	Customer's email address	Maximum 64 characters
firmaadi	BillTo company name	Maximum 255 characters
Faturafirma	BillTo name/surname	Maximum 255 characters
tel	BillTo company Phone	Maximum 32 characters
Fadres	BillTo address line 1	Maximum 255 characters
Fadres2	BillTo address line 2	Maximum 255 characters
Filce	BillTo city	Maximum 64 characters
Fil	BillTo state/province	Maximum 32 characters
Fpostakodu	BillTo postal code	Maximum 32 characters
Fulkekodu	BillTo country code	Maximum 3 characters
NakliyeFirma	ShipTo company	Maximum 255 characters
tismi	ShipTo name	Maximum 255 characters
tadres	ShipTo address line 1	Maximum 255 characters
tadres2	ShipTo address line 2	Maximum 255 characters
tilce	ShipTo city	Maximum 64 characters

til	ShipTo state/province	Maximum 32 characters
tpostakodu	ShipTo postal code	Maximum 32 characters
tulkekod	ShipTo country code	Maximum 3 characters
idl	Id of item #l, required for item #l	Maximum 128 characters
itemnumberl	Item number of item #l	Maximum 128 characters
productcodel	Product code of item #l	Maximum 64 characters
qtyl	Quantity of item #l	Maximum 32 characters
descl	Description of item #l	Maximum 128 characters
pricel	Price of item #l	Maximum 32 characters
amount	Subtotal of item #l	Maximum 32 characters

Transaction Response Parameters

Parameter	Description	Format
AuthCode	Transaction Verification/Approval/Authorization code	6 characters
Response	Payment status	Possible values: "Approved", "Error", "Declined"
HostRefNum	Host reference number	12 characters
ProcReturnCode	Transaction status code	2 digits, "00" for authorized transactions, "99" for Nestpay errors, others for ISO-8583 error codes
TransId	Nestpay Transaction Id	Maximum 64 characters
ErrMsg	Error message	Maximum 255 characters
ClientIp	IP address of the customer	Maximum 15 characters formatted as "###.###.###.###"
ReturnOid	Returned order ID, must be same as input orderId	Maximum 64 characters
MaskedPan	Masked credit card number	12 characters, XXXXXX***XXX
EXTRA.TRXDATE	Transaction Date	17 characters, formatted as "yyyyMMdd HH:mm:ss"
rnd	Random string, will be used for hash comparison	Fixed length, 20 characters
HASH	Hash value	Fixed length, 20 characters

MPI Response Parameters

Parameter	Description	Format
mdStatus	Status code for the 3D transaction	1=authenticated transaction 2, 3, 4 = Card not participating or attempt 5,6,7,8 = Authentication not available or system error 0 = Authentication failed
merchantID	MPI merchant ID	15 characters
txstatus	3D status for archival	Possible values "A", "N", "Y"
iReqCode	Code provided by ACS indicating data that is formatted correctly, but which invalidates the request. This element is included when business processing cannot be performed for some reason.	2 digits, numeric
iReqDetail	May identify the specific data elements that caused the Invalid Request Code (so never supplied if Invalid Request Code is omitted).	
vendorCode	Error message describing *iReqDetail* error.	
PAResSyntaxOK	If PARes validation is syntactically correct, the value is true. Otherwise value is false.	"Y" or "N"
ParesVerified	If signature validation of the return "N" message is successful, the value is true. If PARes message is not received or signature validation fails, the value is false.	"Y" or
eci	Electronic Commerce Indicator	2 digits, empty for non-3D transactions
cavv	Cardholder Authentication Verification Value, determined by ACS.	28 characters, contains a 20 byte value that has been Base64 encoded, giving a 28 byte result.
xid	Unique internet transaction ID	28 characters, base64 encoded
cavvAlgorthm	CAVV algorithm	Possible values "0", "1", "2", "3"
md	MPI data replacing card number	Alpha-numeric
Version	MPI version information	3 characters l(ike "2.0")
sID	Schema ID	"1" for Visa, "2" for Mastercard

MdErrorMsg	Error Message from MPI (if any)	Maximum 512 characters