NestPay®

Merchant Integration
3D Pay Hosting

Document Information

Project/Product Name			
Project Manager			
Document Version No			
Document Code			
Prepared By		**Preparation Date**	22/10/2019
Reviewed By		**Review Date**	22/10/2019

Distribution List

From	**Date**	**Phone/Fax**

To	**Action***	**Due Date**	**Phone/Fax**

* Action Types: Approve, Review, Inform, File, Action Required, Attend Meeting, Other (please specify)

Version History

Ver. No.	Ver. Date	Prepared	Explanation
1.0	09.05.2019	Algı Acarbay	HTTP parameters for 3DS 2.0 optional parameters included
1.1	15.05.2019	Algı Acarbay	Workflow added, input parameters updated, output parameters added
1.2	11.11.2019	Damlanur Keskin Çantalı	Example of HTTP Request and Response forms added
1.3.	12.11.2019	Damlanur Keskin Çantalı	Examples of MPI Messages, 3D Fields are sent to Acquire Banks and Nestpay Screens related with 3D authentication added
1.4.	10.02.2020	Gökçe Ziya	3DS 2.0 and 1.0 differences and comparison for Merchant side. 3DS 2.0 messages (AReq/ARes, CReq/Cres) has removed.

Proprietary Notice

The information contained in all sheets of this document, proposal, or quotation constitutes trade secrets and/or information that is commercial or financial and is deemed confidential or privileged. It is furnished to prospective customer in confidence with the understanding that prospective customer will not, without the permission of Asseco-SEE Teknoloji A.Ş. (from now on called Asseco-SEE or Asseco), use or disclose for other than evaluation purposes the information contained herein which is solely confidential and proprietary to Asseco-SEE ("***Asseco-See Confidential Information***"). In the event a contract is awarded on the basis of this document, proposal, or quotation, prospective customer shall have the right to use and disclose Asseco-See Confidential Information to the extent provided in the contract. The restriction does not limit prospective customer right to use or disclose such information if obtained from another source without restriction.

Contents

1. 3D Pay Hosting Model

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

Basic Features:

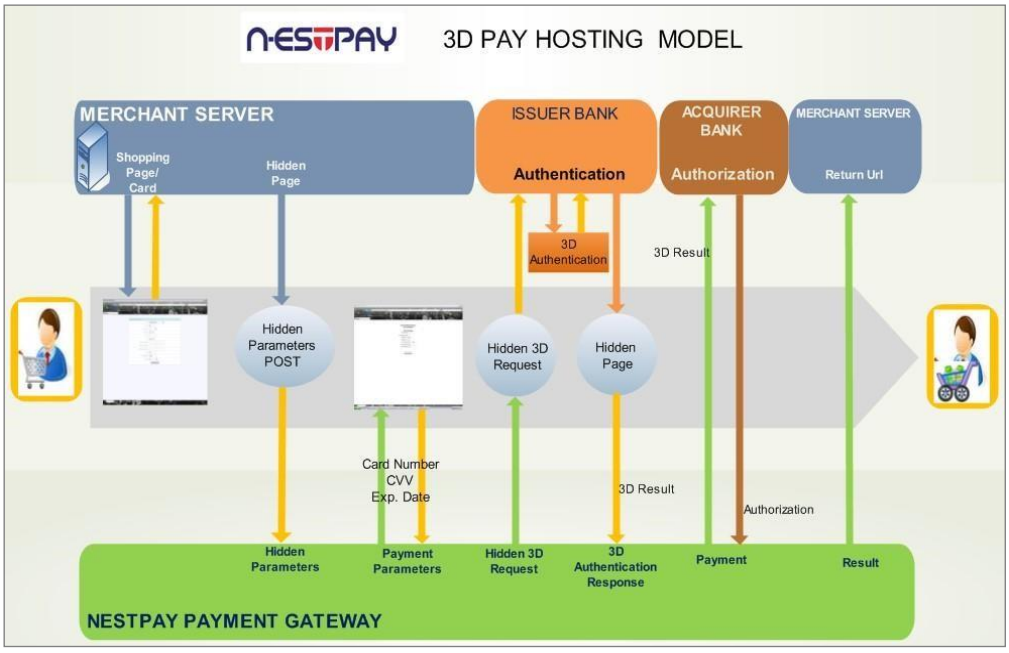
- Enables processing of 3D secure card transactions.
- HTTP Post method is supported for merchant integration.
- Payment is done automatically by NestPay.

After obtaining all necessary shopping data from the customer like order amount, currency, customer name/surname etc., merchant server generates a unique order ID. Necessary parameters are posted using HTTP Post method to NestPay Payment Gateway.

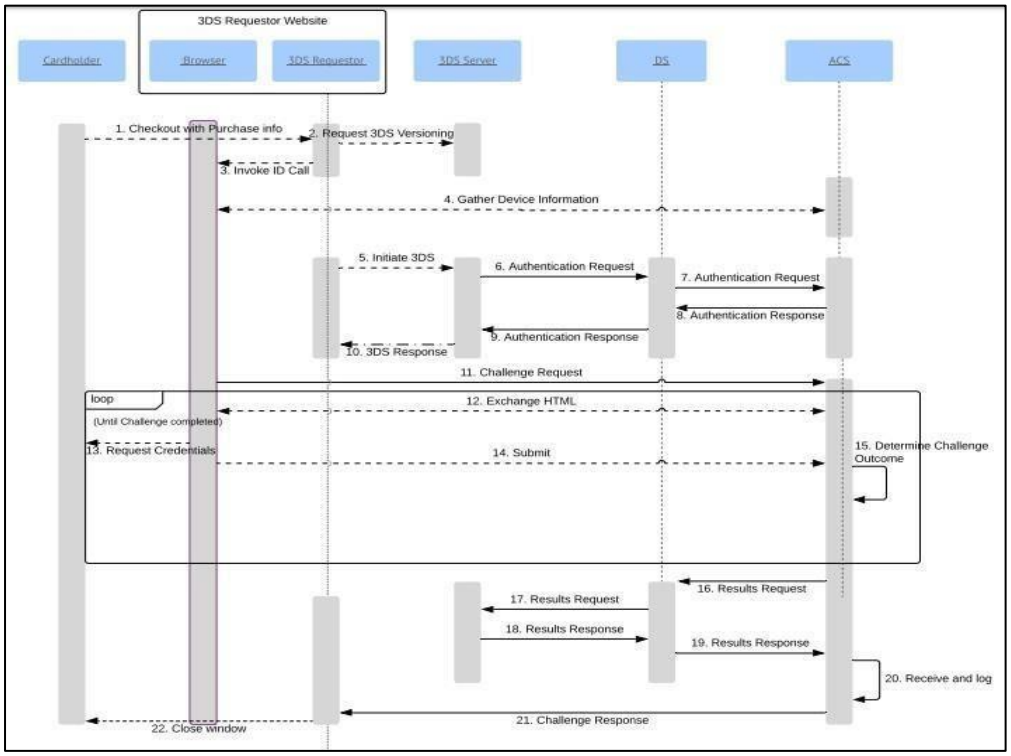
For credit card payment methods like Visa, MasterCard, etc., the merchant server needs to submit card details like card number, CVV2, and expiry date information. After the order/card data is obtained from the user, 3D flow including enrolment and authentication queries starts. In 3D flow, 3D authentication 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 include using 3D secure passwords, one-time passwords, and security questions.

Using this model,

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



3DPayHosting Model Diagram



3D PayHosting2.0 Model Diagram (Browser Based)

2. Quick Start Guide

This section will describe how to perform a successful Sale VISA transaction with **3D Pay Hosting Model**.

2.1 Generate Hash for Client Authentication

Hash ver3 is the recommended hash calculation method. Please see Hash ver3 documentation for details. Hash ver2 is descibed below and is supported for backward compatibility.

2.1.1 Hash Version 2 – SHA 512 Algorithm

Hash for Version 2 is the base64-encoded version of the hashed text which is generated with SHA512 algorithm. For using Hash Version 2, "***hashAlgorithm"*** parameter should be sent in the request with the value of **"ver2"**.

To generate the hash for client authentication, the following parameter values should be appended in the order given below by using pipeline "|" as a separator. If a parameter is to be skipped, a double pipeline "||" can be used instead.

```
plaintext = clientid + | + oid + | + amount +| + okurl +| + failurl + | + transaction type + | + instalment
    + | + rnd +|||| + currency +| + storeKey
```

Note: In case "|" character needs to be used in a parameter, "\" character can be used for escaping. Additionally, if the "\" character is also used in a parameter, we use another "\" character before and then append it to hash plain text. An example is shown below:

```
Original Text  : ORDER-256712jbs\j6b|
Escaped Text : ORDER-256712jbs\\j6b\|
```

Given parameters

Clienid : 990000000000001 **oid**
: ORDER256712jbs\j6b|
amount : 91.96
okurl : https://www.teststore.com/success.php
failurl : https://www.teststore.com/fail.php
transaction type : Auth **instalment** : 2
rnd : asdf
currency : 949
storekey : AB123456\|

- **Hash**

Plaintext= 990000000000001|ORDER256712jbs\\j6b\||91.96|https://www.teststore.com/success.php|https://www.teststore.com/fail.php|Auth|2|asdf||||949|AB123456\\\| Hash = Base64(SHA512(plaintext))

2.2 Posting Hidden Parameters

clientid : Merchant ID (given by Nestpay)

storetype : **3d_pay_hosting**

hash : Hash value for client authentication

trantype : "Auth"

amount : transaction amount

currency : ISO code of transaction currency (949 for TL)

oid : Unique identifier of the order

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)

encoding : Page encoding

Sample HTTP form with mandatory parameter set

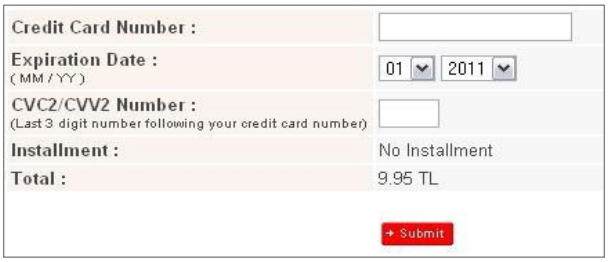
```
<form method="post" action="https://host/fim/est3dgate">

<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="trantype" 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="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" name="encoding" value="utf-8" />

</form>
```

2.3 Payment Page

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



2.4 3D Authentication

In 3D flow, 3D authentication information of the customer is collected by the issuer bank. The methods for 3D authentication can be different for different issuers. Examples of 3D authentication methods include using 3D secure passwords, one-time passwords, and security questions.

2.5 Transaction Result Page

The transaction result will be displayed to the customer. If the transaction is successful, the authorization code will be displayed. The customer will be redirected to *okUrl* if *refreshtime* has passed.

2.6 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 the merchant are returned back to the merchant. In addition to merchant parameters, gateway returns the transaction response parameters and MPI response parameters related to 3D secure transaction flow, which are listed in Appendix A.

Basic transaction response parameters for fully 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, transaction response parameters would be:

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

3. Integration Basics

3.1 HTTP Post Integration

After receiving a valid order, parameters are posted to NestPay Payment Gateway as hidden parameters with the HTTP form. In addition to mandatory parameters, the merchant can post order billing/shipping and order item details to payment gateway, which can be viewed later on Merchant Administration Panel. For optional parameter 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 NestPay system.

3.1.1 Sample HTTP form with mandatory and optional parameters

```
<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="trantype" value="Auth" />
    <input type="hidden" name="amount" value="91.96" />
    <input type="hidden" name="currency" value="949" />
<input type="hidden" name="instalment" value="">
    <input type="hidden" name="oid" value="1291899411421" />
    <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" name="hashAlgorithm" value="ver2">

Optional parameters
    <input type="hidden" name="tel" value="012345678">
    <input type="hidden" name="email" value="test@test.com">

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

        <input type="hidden" name="printBillTo" value="true">
        <input type="hidden" name="BillToCompany" value="Billing Company">
        <input type="hidden" name="BillToName" value="Bill John Doe">
                <input type="hidden" name="BillToStreet1" value="Address line 1">
                <input type="hidden" name="BillToStreet2" value="Address line 2">
        <input type="hidden" name="BillToStreet3" value="Address line 3">



        <input type="hidden" name="BillToCity" value="Istanbul">
        <input type="hidden" name="BillToStateProv" value="mystate">
        <input type="hidden" name="BillToPostalCode" value="12345">
        <input type="hidden" name="BillToCountry" value="616">

<!-- Shipping Parameters [All Optional]-->
        <input type="hidden" name="printShipTo" value="true">
        <input type="hidden" name="ShipToCompany" value="Shipping Company">
        <input type="hidden" name="ShipToName" value="Ship John Doe">
        <input type="hidden" name="ShipToStreet1" value="Address line 1">
        <input type="hidden" name="ShipToStreet2" value="Address line 2">
        <input type="hidden" name="ShipToStreet3" value="Address line 3">
        <input type="hidden" name="ShipToCity" value="Istanbul">
        <input type="hidden" name="ShipToStateProv" value="mystate">

        <input type="hidden" name="ShipToPostalCode" value="12345">
        <input type="hidden" name="ShipToCountry" value="616">
```

```
<!-- Order Item Parameters [All Optional]-->
        <input type="hidden" name="ItemNumber1" value="a5">
        <input type="hidden" name="ProductCode1" value="a5">
        <input type="hidden" name="Qty1" value="3">
        <input type="hidden" name="Desc1" value="a5 desc">
        <input type="hidden" name="Id1" value="a5">
        <input type="hidden" name="Price1" value="6.25">
       <input type="hidden" name="Total1" value="7.50">


<!—Recurring Payment [All Optional]-->
        <input type="hidden" name="RecurringPaymentNumber" value="6">
        <input type="hidden" name="RecurringFrequencyUnit" value="M">
        <input type="hidden" name="RecurringFrequency" value="1">


</form>
```

3.2 Card Transactions

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

3.2.1 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

3.2.1.1 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

3.2.1.2 Successful Transaction

The authorization code will be displayed. The customer will be redirected to **okUrl** of the merchant server when refresh time has passed. All input parameters along with transaction response parameters will be posted to **okUrl**, and the Response parameter will be "**Approved**"

3.2.1.3 Failed Transaction

The failure message will be displayed. The customer will be redirected to **failUrl** of the merchant server when refresh time has passed. All input parameters along with transaction response parameters will be posted to **failUrl**, and the Response parameter will be "**Declined**" or "**Error**".

3.2.1.4 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
EXTRA.CARDBRAND : Brand of Credit Card (Visa, MasterCard, etc…) **rnd**
: Random string, will be used for hash comparison
HASHPARAMS : Contains the field names used for hash calculation. Field names are appended with ":" character

HASHPARAMSVAL : Contains the appended hash field values for hash calculation. Field values appended with the same order in *HASHPARAMS* field

HASH : Hash value of *HASHPARAMSVAL* and merchant password field

3.2.1.5 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

3.2.1.6 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 than "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 detailed 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 the acquirer authorization step. *ErrMsg* parameter will give the detailed description of the error.

3.3 Hash Checking

After merchant receives the parameters, a hash check needs to be done at merchant's server for validating the parameters. Hash checking ensures that the message is sent by Nestpay only. **"*hashAlgorithm*"** parameter value should be set as **"ver3"** so that Hash Version 3 will be selected as hash calculation method. Please see Hash Version 3 documentation for details.

Hash Version 2 is supported for backward compatibility. For using Hash Version 2, **"*hashAlgorithm*"** parameter value should be set as **"ver2".**

3.3.1 Generating the plain text for Hash Ver-2

The parameters used for hash calculation are the following: *clientid, oid, AuthCode, ProcReturnCode, Response, rnd, md, eci, cavv, mdStatus.* Depending on the type of transaction a subset of these parameters will be included in the hash generation:

- HASHPARAMS, HASHPARAMSVAL
- Non 3D-secure card transactions

clientid, oid, AuthCode, ProcReturnCode, Response, rnd

- 3D secure card transactions clientid, oid, AuthCode, ProcReturnCode, Response, mdStatusi eci, cavv ,md, rnd

All the values corresponding to these parameters are appended in this specific order.

Note: In generation of the hash, pipeline "|" character is used as a separator between parameters. The resulting string will be the same as *HASHPARAMSVAL* parameter values. The merchant password is appended as the final value to the end of this string. The resulting hash is the base64-encoded version of the hashed text which is generated with SHA-512 algorithm. Under normal conditions, generated hash text must be the same as *HASH* parameter value posted by NestPay payment gateway. If not, merchant should contact to Nestpay support team.

Note: In case "|" character needs to be used in a parameter, "\" character can be used for escaping. Additionally, if the "\" character is also used in a parameter, we use another "\" character before it and then append it to hash plain text. An example is shown below:

Original order id : ORDER-256712jbs\j6b|
Escaped order id : ORDER-256712jbs\\j6b\|

Example: Non 3D card transaction

Assuming that the transaction response parameters are:

clientid, oid, AuthCode, ProcReturnCode, Response, rnd

clientid : 990000000000001
oid : ORDER256712jbs\j6b|
AuthCode : 213216
ProcReturnCode : 540000 **Response** : Approved
rnd : LdXGnw20ZupyXVr1XCu2
storeKey : AB123456\|

HASHPARAMSVAL :
990000000000001|ORDER256712jbs\\j6b\||213216|540000|Approved|LdXGnw20ZupyXVr1XCu2
HASHPARAMS : clientid|oid|AuthCode|ProcReturnCode|Response|rnd
HASH :
UfrEVCY1lRoThXE571XkSK8a/LnoJOCLgijId+by7qGLKlyChhgnbwRwEJrFQSpSacH9DYk0Zkup2cmyFXoDLg==

The merchant hash text will be generated with clientid, oid, ProcReturnCode, Response, rnd (and store key of the merchant as secret hash element).

And the merchant hash is based64-encoded(SHA512(plain)). The result hash should match the returning parameter *HASH.*

Note: Merchant should check Hash parameter of HASHPARAMS & HASHPARAMSVAL return values by the bank.

3.3.2 Security Notes

Merchant must check the mandatory parameters in **HASHPARAMS** parameter. For the approved provisions, *ClientId*, *OrderId* and *Response* parameter names must be present in **HASHPARAMS.** Merchant must also check the values of these mandatory return parameters. *ClientId* must be the clientId assigned to the merchant.

If these parameters are not present or do not have valid values, merchant should not process this transaction and must report it to the Support Desk immediately.

4. Dual Message with Auto Capture/Void by Callback (Confirmation)

The merchant starts payment flow by sending a PreAuth request to NestPay with parameter values trantype=PreAuth and CallbackResponse=true. In addition to the required parameters, CallbackUrl should be sent so that NestPay can send the confirmation callback to the merchant. Upon receiving confirmation callback, the merchant is expected to send callback response to NestPay.

The following callback responses will be supported:

- **Acknowledgement** : Merchant's response should be "APPROVED".
  The transaction is acknowledged by the merchant. The customer is asked to check the status of his/her order on the merchant's site. The merchant needs to manage capture or void manually via API or Merhant Center interface.

- **Acknowledgement&Settle**: Merchant's response should be "ACTION=POSTAUTH".
  Optionally, the merchant can send BUSINESSDATE callback response parameter, for example, "ACTION=POSTAUTH&BUSINESSDATE=04/11/2015".
  In this case, an automatic capture (PostAuth) request will be sent to the acquirer bank. BUSINESSDATE value will be used to define the time of the capture request if financial time is not defined for the merchant.

  The merchant can send up to 10 extra parameters, DIMCRITERIA1, …, DIMCRITERIA10 as part of callback response, or example,
  "ACTION=POSTAUTH&BUSINESSDATE=04/11/2015&DIMCRITERIA1=VAL1&DIMCRITERIA2=VAL2&DIMCRITERIA3=VAL3"

- **Acknowlegement&Void**: Merchant's response should be "ACTION=VOID".
  The transaction is cancelled by the merchant. A VOID request will automatically be sent to the acquirer bank and the customer will not be debited.

  The merchant can send up to 10 extra parameters, DIMCRITERIA1, …, DIMCRITERIA10 as part of callback response, or example,
  "ACTION=VOID&DIMCRITERIA1=VAL1&DIMCRITERIA2=VAL2&DIMCRITERIA3=VAL3"

- **Timeout:** If timeout occurs with the merchant's web site, the customer is asked to check the status of his/her order on the merchant's site. In this case, an email notification will be sent to CMI and the merchant. Merchant needs to manage capture or void manually via API or Merhant Center interface.

- **Merchant's reponse syntax error:** If there is an error from the merchant's web site, the customer is asked to check the status of his/her order in the merchant's site. Merchant needs to manage capture or void manually via API or Merhant Center interface.

Additionally, NestPay supports to receive up to ten fields in callback response from merchants. The Merchant should manage and reserve specific fields for his purpose. Merchant shouldn't use the same field both in payment request and callback response. If the merchant sends a field in

callback response which he already sent in payment request, nestpay will ignore the value of the field in callback response.

5. Code Samples

In this section, code samples for merchants using 3D Pay Hosting Model are provided.

5.1 ASP Code Sample

5.1.1 Request Sample Codes

5.1.1.1 Hash Version 2

For ASP Classic, there is no built-in SHA-512 Hash Calculation library. Because of this reason, ASP classic code should be extended to use C# or VB.Net code which have a built-in library for SHA 512 hash generation. Below, you can find example C# and VB Code example for Hash Version 2.

5.1.2 Response Code Sample

```
<html>
<head>
<title>3D Pay Payment Page</title>
  <meta http-equiv="Content-Language" content="tr">
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-9">
  <meta http-equiv="Pragma" content="no-cache">
  <meta http-equiv="Expires" content="now">


</head>
<body>
<!-- #include file = "hex_sha1_js.asp" -->
 <h1>Payment Page</h1>
   <h3> Payment Response</h3>
   <table border="1">
      <tr>
         <td><b>Parameter Name</b></td>
         <td><b>Parameter Value</b></td>
      </tr>


    dim
    obj,ok,mdstatus,hashparams,hashparamsval,hash,index1,index2,storekey,hashparam,val,hashval,par
    amsval
        dim paymentparams(5)
        ok = 1

'hash checking parameters
        storekey = "xxxxxx"
        index1 = 1      index2
= 1
        hashparams = request.form("HASHPARAMS")
        hashparamsval =
request.form("HASHPARAMSVAL")       hashparam =
request.form("HASH")            paramsval = ""
        paymentparams(0) = "AuthCode"
        paymentparams(1) = "Response"
        paymentparams(2) = "HostRefNum"
        paymentparams(3) = "ProcReturnCode"
        paymentparams(4) = "TransId"
```

```
                                        paymentparams(5) = "ErrMsg"

            for each obj in request.form
        ok = 1
                                    for each item in paymentparams
                                    if(item = obj) Then

                ok = 0
                        exit for
            end if
        next
                    if ok = 1 then
            response.write("<tr><td>"&obj & "</td><td>" & request.form(obj) & "</td></tr>")
            end if   next

                    </table>
              <br>
              <br>

'hash cheking

        while index1 < Len(hashparams)
        index2 = InStr(index1,hashparams,":")
        xvalx = Mid(hashparams,index1,index2 - index1)
            val = request.form(xvalx)                if
val = null then                 val = ""
        end if                  paramsval = paramsval
& val           index1 = index2 + 1
                Wend
        hashval = paramsval & storekey
        hash = b64_sha1(hashval)
    'response.write("hash=" & hash & "<br/>hashparam=" & hashparam &"<br>paramsval=" &
    paramsval &"<br>hashparamsval=" & hashparamsval )


                                                                    if hash <> hashparam or
paramsval <> hashparamsval then
                                                        response.write("<h4>Security Alert. The digital
signature is not
valid.</h4>")   end if
                                    mdstatus = Request.Form("mdStatus")
            if mdstatus = 1 or mdstatus = 2  or mdstatus = 3 or mdstatus = 4 Then
```

```
<h5>3D Transaction is Success</h5><br/>
            <h3> Payment Host</h3>
            <table border="1">
               <tr>
                  <td><b>Parameter Name</b></td>
                  <td><b>Parameter Value</b></td>
               </tr>

    for each item in paymentparams
 response.Write("<tr><td>" & item &"</td><td>" & request.form(item) & "</td></tr>")  next  </table>
if "Approved" = request.form("Response") Then
        Response.write("<h6>Transaction is Success</h6>")
    Else
                Response.write("<h6>Transaction is not Success</h6>")
        end if
    else
                Response.Write("<h6>3D not Approved </h6>")
        end if </body>
</html>
```

5.2 .Net Code Sample

5.2.1 Request Sample Codes

5.2.1.1 Hash Version 2

5.2.1.1.1 .Net - C# Code Sample

```
<%@ page language="C#" autoeventwireup="true"
        inherits="_3DModel, App_Web_fr4klrwv"%>



<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>3D Model</title>
```

```
</head>
<body>
                    <%
    // plaintext = clientid + | + oid + | + amount +| + okurl +| + failurl + | + transaction type + | +
     instalment +
                                                                    // | + rnd +|||| + currency +| +
storeKey


                                        //unEscaped values
                                                                    String orgClientId =
"990000000000001";
                                                                    String orgOid =
"ORDER256712jbs\\j6b|";
                                                            String orgAmount = "91.96";
                                                                            String orgOkUrl =
"https://www.teststore.com/success.php";
                                                                        String orgFailUrl =
"https://www.teststore.com/fail.php";
                                                                String orgTransactionType = "Auth";
                                                            String orgInstallment = "";
                                                                    String orgRnd =
DateTime.Now.ToString();
                                                            String orgCurrency = "949";
                                         // escaped values
                                                                        String clientId =
orgClientId.Replace("\\",
"\\\\").Replace("|", "\\|");

                                                                            String oid =
orgOid.Replace("\\", "\\\\").Replace("|", "\\|");
                                                                            String amount =
orgAmount.Replace("\\",
"\\\\").Replace("|", "\\|");
                                                                          String okUrl =
orgOkUrl.Replace("\\", "\\\\").Replace("|",
"\\|");
                                                                          String failUrl =
orgFailUrl.Replace("\\", "\\\\").Replace("|",
"\\|");
                                                                        String transactionType =
orgTransactionType.Replace("\\", "\\\\").Replace("|", "\\|");
                                                                            String installment =
orgInstallment.Replace("\\",
"\\\\").Replace("|", "\\|");

                                                                            String rnd =
orgRnd.Replace("\\", "\\\\").Replace("|", "\\|");
                                                                            String currency =
orgCurrency.Replace("\\",
"\\\\").Replace("|", "\\|");
                                                                            String storeKey =
"AB123456\\|".Replace("\\",
"\\\\").Replace("|", "\\|");


                    String plainText = clientId + "|" + oid + "|" + amount + "|" + okUrl + "|" + failUrl +
"|"
```

```
                                                                                +
transactionType + "|" + installment +
"|" + rnd + "||||" +currency + "|" + storeKey;



System.Security.Cryptography.SHA512 sha = new
    System.Security.Cryptography.SHA512CryptoServiceProvider();      byte[] hashbytes =
System.Text.Encoding.GetEncoding("ISO-8859-9").GetBytes(plainText);            byte[] inputbytes
= sha.ComputeHash(hashbytes);
```

```
                                                                String hash =
Convert.ToBase64String(inputbytes);

                                        String description = "";
                                    String xid = "";
                                    String lang = "";
            String email = "";
String userid = "";
%>
                <center>
                                            <form method="post"
action="https://<Host_Address>/<3dgate_path>">
                        <table>
                                <tr>
                                                <td>Credit Card Number</td>

                                                        <td><input type="text" name="pan"
size="20" />
                                </tr>
                                <tr>
                                            <td>CVV</td>
                                                                <td><input type="text"
name="cv2" size="4" value="" /></td>
                                </tr>
                                <tr>
                                                <td>Expiration Date Year</td>
                                        <td><input type="text"
name="Ecom_Payment_Card_ExpDate_Year"                                                   value=""
/></td>
                                </tr>
                                <tr>
                                                        <td>Expiration Date Month</td>
                                                <td><input type="text"

name="Ecom_Payment_Card_ExpDate_Month value=" "/></td>
                                </tr>
                                <tr>
                                                <td>Choosing Visa Master Card/td>
                                                        <td><select name="cardType">
                                                                        <option
value="1">Visa</option>
                                                                        <option
value="2">MasterCard</option>
                                            </select>
                                </tr>
                                <tr>
                                        <td align="center" colspan="2"><input type="submit"
                                        value="Complete Payment" /></td>
                                </tr>
                        </table>
```

<input type="hidden" name="clientid" value="<%=orgClientId%>">

```
                        <input type="hidden" name="amount" value="<%=orgAmount%>">
                        <input type="hidden" name="oid" value="<%=orgOid%>">
                        <input type="hidden" name="okUrl" value="<%=orgOkUrl%>">
                        <input type="hidden" name="failUrl" value="<%=orgFailUrl%>">
                        <input type="hidden" name="TranType" value="<%=orgTransactionType%>">
                <input type="hidden" name="Instalment" value="<%=orgInstallment%>"> <input
type="hidden" name="currency" value="<%=orgCurrency%>">
<input type="hidden" name="rnd" value="<%=orgRnd%>">
<input type="hidden" name="hash" value="<%=hash%>">
<input type="hidden" name="storetype" value="3D_PAY_HOSTING">
<input type="hidden" name="lang" value="tr">
<input type="hidden" name="hashAlgorithm" value="ver2"> </form>
    </center>
</body>
</html>
```

5.2.1.1.2 .Net - VB Code Sample

```
<%@ page language="VB" autoeventwireup="true"
        inherits="_3DModel, App_Web_fr4klrwv"%>


<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">


<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>3D Model</title>
</head>
<body>
    <%
                    Dim orgClientId As String = "990000000000001"
                        Dim orgOid As String = "ORDER256712jbs\j6b|"
                        Dim orgAmount As String = "91.96"
                        Dim orgOkUrl As String = "https://www.teststore.com/success.php"
                        Dim orgFailUrl As String = "https://www.teststore.com/fail.php"
                        Dim orgTransactionType As String = "Auth"
                        Dim orgInstallment As String = ""
                        Dim orgRnd As String = DateTime.Now.ToString()


                        Dim orgCurrency As String = "949"


                Dim clientId As String = orgClientId.Replace("\", "\\").Replace("|", "\|")
                        Dim oid  As String = orgOid.Replace("\", "\\").Replace("|", "\|")
                        Dim amount As String  = orgAmount.Replace("\", "\\").Replace("|", "\|")
                        Dim okUrl As String = orgOkUrl.Replace("\", "\\").Replace("|", "\|")
                        Dim failUrl As String = orgFailUrl.Replace("\", "\\").Replace("|", "\|")

Dim transactionType As String = orgTransactionType.Replace("\",  "\\").Replace("|",
    "\|")
                                    Dim installment As String = orgInstallment.Replace("\",
"\\").Replace("|", "\|")                                        Dim rnd As String = orgRnd.Replace("\",
"\\").Replace("|", "\|")


                        Dim currency As String = orgCurrency.Replace("\", "\\").Replace("|", "\|")
                        Dim storeKey As String = "AB123456\|".Replace("\", "\\").Replace("|", "\|")


                                    Dim plainText As String  = clientId + "|" + oid + "|" + amount + "|" +
   okUrl + "|" + failUrl + "|" + transactionType + "|" + installment + "|" + rnd + "||||" + currency + "|"
    + storeKey
                Dim result As Byte()
```

```
            Dim mixer As String
            Dim sha As New System.Security.Cryptography.SHA512Managed()


            result = sha.ComputeHash(System.Text.Encoding.ASCII.GetBytes(plainText))
              Dim hashValue As String  = Convert.ToBase64String(result)
              Dim description As String = "";
              Dim xid As String = "";
              Dim lang As String = "";
              Dim email As String = "";
              Dim userid As String = "";
    %>
    <center>
        <form method="post" action="https://<Host_Address>/<3dgate_path>">
                <table>
                        <tr>
                                <td>Credit Card Number</td>
                                <td><input type="text" name="pan" size="20" />
                        </tr>
                        <tr>
                                <td>CVV</td>
                                <td><input type="text" name="cv2" size="4" value="" /></td>
                        </tr>
                        <tr>
                                <td>Expiration Date Year</td>
                                <td><input type="text" name="Ecom_Payment_Card_ExpDate_Year"
                                        value="" /></td>
                        </tr>
                        <tr>
                                <td>Expiration Date Month</td>
                                        <td><input type="text"




name="Ecom_Payment_Card_ExpDate_Month value=" "/></td>
                                </tr>
                                <tr>
                                                        <td>Choosing Visa Master Card/td>
                                                                <td><select name="cardType">
                                                                        <option
value="1">Visa</option>
                                                                        <option
value="2">MasterCard</option>
                                                </select>
                                </tr>
                                <tr>
                                        <td align="center" colspan="2"><input type="submit"
                                        value="Complete Payment" /></td>
                        </tr>
                </table>
```

```
                <input type="hidden" name="clientid" value="<%=orgClientId%>">
                        <input type="hidden" name="amount" value="<%=orgAmount%>">
                        <input type="hidden" name="oid" value="<%=orgOid%>">
                        <input type="hidden" name="okUrl" value="<%=orgOkUrl%>">
                        <input type="hidden" name="failUrl" value="<%=orgFailUrl%>">
                        <input type="hidden" name="TranType" value="<%=orgTransactionType%>">
                        <input type="hidden" name="Instalment" value="<%=orgInstallment%>">


                        <input type="hidden" name="currency" value="<%=orgCurrency%>">
                        <input type="hidden" name="rnd" value="<%=orgRnd%>">
                        <input type="hidden" name="hash" value="<%=hash%>">
                        <input type="hidden" name="storetype" value="3D_PAY_HOSTING">
                        <input type="hidden" name="lang" value="tr">
                        <input type="hidden" name="hashAlgorithm" value="ver2">
        </form>
    </center>
</body>
</html>
```

5.2.2 Response Code Sample

5.2.2.1 .Net - C# Sample Code

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>3d Pay Payment Page</title>
</head>
<body>
<h1>3D Payment Page</h1>
   <h3>Payment Response</h3>
   <table border="1">
       <%
   String originalClientId = "xxxxxx";
        String [] mustParameters = new String[] {"clientid","oid","Response"};
boolean isValid = true;

      for(int i=0;i<mustParameters.length;i++)
{
        if(Request.Form[mustParameters[i]] == null || Request.Form[mustParameters[i]] == "" )
         {
                  if(mustParameters[i].equals("oid")){
            if(Request.Form["ReturnOid"] == null || Request.Form["ReturnOid"] == "" ){
isValid = false;
```

```
                Response.Write("<tr><td>Missing Required Param</td>"+"<td>oid /
    ReturnOid</td></tr>");

              }
            }else{
                isValid = false;
                    Response.Write("<tr><td>Missing Required
    Param</td>"+"<td>"+mustParameters[i]+"</td></tr>");
            }
           }
         }
       if(!Request.Form.Get("clientid").Equals(originalClientId)){
          Response.Write("<h4>Security Alert. Incorrect Client Id.</h4>");          return;

       }
if(!isValid){
          Response.Write("<h4>Security Alert. The digital signature is not valid. Required Paramaters are
    missing.</h4>");

          return;
       } else {


    %>
<tr>


<td><b>Parameter Name</b></td>
                <td><b>Parameter Value</b></td>
        </tr>
        <%
```

```
String[]paymentparams = new String[] { "AuthCode", "Response", "HostRefNum",  "ProcReturnCode",
                                        "TransId", "ErrMsg" };
                    IEnumerator e =
Request.Form.GetEnumerator();                       while
(e.MoveNext()) {
                        String xkey = (String)
e.Current;                          String xval =
Request.Form.Get(xkey);                           boolean ok
= true;
                        for (int i = 0; i < paymentparams.Length;
i++) {                              if
(xkey.Equals(paymentparams[i])) {                               ok
= false;                                  break;


}
}
if (ok)
                                                                    Response.Write("<tr><td>" + xkey +
"</td><td>" + xval + "</td></tr>");
                    }
                %>
                </table>


        <%
                                        String hashparams = Request.Form["HASHPARAMS"];
                                        String hashparamsval = Request.Form["HASHPARAMSVAL"];
                        String hash = "";
                            String storekey = "xxxxxx";
            String paramsval = "";
                String hashval =
"";         int index1 = 0, index2
= 0;


                if
(Request.Form.Get("hashAlgorithm").Equals("ver2")){
                        string[] parsedParams =


hashparams.Split('|');                      foreach (string parsedParam in parsedParams)
                        {
                                String val = Request.Form.Get(parsedParam) == null ? "" :
```

```
    Request.Form.Get(parsedParam);                                  paramsval +=
val.Replace("\\", "\\\\").Replace("|", "\\|") + "|";
                                    }


                                                                    hashval = paramsval +
storekey.Replace("\\", "\\\\").Replace("|",
"\\|");                    String hashparam = Request.Form.Get("HASH");


System.Security.Cryptography.SHA512 sha = new
    System.Security.Cryptography.SHA512CryptoServiceProvider();   byte[] hashbytes =
System.Text.Encoding.GetEncoding("ISO-8859-9").GetBytes(hashval);
         byte[] inputbytes = sha.ComputeHash(hashbytes);                        hash =
Convert.ToBase64String(inputbytes);
                } else {
                          do
                          {
                                                  index2 = hashparams.IndexOf(":", index1);
                                                                  String val =
Request.Form.Get(hashparams.Substring(index1, index2-index1))
    ==                                    null ? "" : Request.Form.Get(hashparams.Substring(index1,
index2-index1));        paramsval += val;        index1 = index2 + 1;
                          }
                                        while (index1 < hashparams.Length);

                                        hashval = paramsval + storekey;
                                                  String hashparam = Request.Form.Get("HASH");

                                                  System.Security.Cryptography.SHA1 sha = new
    System.Security.Cryptography.SHA1CryptoServiceProvider();
         byte[] hashbytes = System.Text.Encoding.GetEncoding("ISO-8859-
    9").GetBytes(hashval);                    byte[] inputbytes
= sha.ComputeHash(hashbytes);

                                                  hash = Convert.ToBase64String(inputbytes);
                }
                                                                    if
(!paramsval.Equals(hashparamsval) ||
!hash.Equals(hashparam)) {
                                                        Response.Write("<h4>Security Alert. The
digital signature is not valid.</h4>");
                                                  Response.Write("<h4>Generated Hash Val : " +
paramsval + "</h4>");
  Response.Write("<h4>Original Hash Val : " + hashparamsval + "</h4>");      }


                                  String mdStatus = Request.Form["mdStatus"];
```

```
            if (mdStatus.Equals("1") || mdStatus.Equals("2") || mdStatus.Equals("3")
|| mdStatus.Equals("4")) { %>
    <h5>3D Transaction is Success</h5>
    <br />
    <h3>Payment Response</h3>
    <table border="1">
        <tr>
    <td><b>Parameter Name</b></td>
     <td><b>Parameter Value</b></td>

        </tr>
<%
            for (int i = 0; i < paymentparams.Length; i++) {
                String paramname = paymentparams [i];
                String paramval = Request.Form.Get(paramname);
                Response.Write("<tr><td>" + paramname + "</td><td>" + paramval +
    "</td></tr>");
            }
        %>
        </table>
<%
        if ("Approved".Equals(Request.Form["Response"])) {
    %>
    <h6>Transaction is Success</h6>
    <%
        } else {
    %>
    <h6>Transaction is not Success</h6>
    <%
        }
        } else {
    %>
<h5>3D Transaction is not Success</h5>
    <%
        }
    }
    %>
</body>
</html>
```

5.2.2.2 .Net - VB.Net Sample Code

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>3d Pay Payment Page</title>
</head>
```

```
<body>
    <h1>3D Payment Page</h1>
    <h3>Payment Response</h3>
<table border="1">


        <%
        Dim originalClientId As String = "xxxxxx"
                Dim mustParameters() As String =
{"clientid","oid","Response"}             Dim isValid As Boolean = True



        For Each paramName As String In mustParameters
                Dim paramValue As String = Request.Form(paramName)
                If String.IsNullOrEmpty(paramValue) Then


                        If paramName.Equals("oid") Then
                          Dim returnOidValue As String = Request.Form("ReturnOid")
            If String.IsNullOrEmpty(returnOidValue) Then                    isValid
= False
                 Response.Write("<tr><td>Missing Required Param</td>"+"<td>oid /
    ReturnOid</td></tr>")
                           End If
            Else
                isValid = false;
             Response.Write("<tr><td>Missing Required Param</td>"+"<td>"+
    paramName+"</td></tr>")
                        End If
      End If
        Next



        If Not Request.Form("clientid").Equals(originalClientId) Then
          Response.Write("<h4>Security Alert. Incorrect Client Id.</h4>")
       End If


       If Not isValid Then
          Response.Write("<h4>Security Alert. The digital signature is not valid. Required Paramaters are
    missing.</h4>")
       Else
      %>
          <tr>
                <td><b>Parameter Name</b></td>
                <td><b>Parameter Value</b></td>
          </tr>
```

```
<%
Dim paymentparams () As String = { "AuthCode", "Response", "HostRefNum", "ProcReturnCode",
  "TransId", "ErrMsg" }
    Dim allKeys() As String = Request.Params.AllKeys


  For Each xKey As String In allKeys
        Dim xval As String = Request.Form(xkey)
        Dim ok As Boolean = True
```

```
                    For Each paymentParam As String In
paymentparams                       If
xkey.Equals(paymentParam) Then                      ok =
False
                                    Exit For
                End If
                    Next

            If ok Then
                    Response.Write("<tr><td>" + xkey + "</td><td>" + xval +
"</td></tr>")                   End If
        Next
        %>
    </table>

    <%
       Dim hashparams As String = Request.Form("HASHPARAMS")
       Dim hashparamsval As String = Request.Form("HASHPARAMSVAL")
       Dim storekey As String = "xxxxxx"
       Dim paramsval As String = ""
       Dim index1 As Integer = 0
       Dim index2 As Integer = 0
        Dim hash As String = ""
        Dim hashparam As String = ""
        Dim hashval As String = ""



        If Request.Form("hashAlgorithm") == "ver2" Then
            Dim parsedParams() As String = hashparams.split("|")

                    For Each parsedParam As String In
parsedParams                        If
String.IsNullOrEmpty(parsedParam) Then
                            val = ""


                        Else
                val = parsedParam
                    End If
```

```
                Next

                        Dim hashval As String = paramsval + storekey.Replace("\", "\\").Replace("|",
"\|")                   Dim hashparam As String = Request.Form("HASH")

                Dim result As Byte()
  Dim sha As New System.Security.Cryptography.SHA512Managed()    result =
sha.ComputeHash(System.Text.Encoding.ASCII.GetBytes(hashval))                              hash
        =
Convert.ToBase64String(result)
        Else
                        Do While index1 < hashparams.Length
                                index2 =
hashparams.IndexOf(":", index1)
   Dim hashedParamValue As String = Request.Form(hashparams.Substring(index1, index2 - index1))
                        Dim val As String
                        If String.IsNullOrEmpty(hashedParamValue) Then
                                        val
= ""                    Else
        val = hashedParamValue
        End If
        paramsval +=
val                             index1 = index2
+ 1
                Loop
                Dim hashval As String = paramsval + storekey
                Dim hashparam As String = Request.Form("HASH")

                Dim result As Byte()
                        Dim sha As New System.Security.Cryptography.SHA512Managed()
                        result =
sha.ComputeHash(System.Text.Encoding.ASCII.GetBytes(hashval))                      hash
As String  = Convert.ToBase64String(result)
        End If
       If (Not paramsval.Equals(hashparamsval)) Or  (Not hash.Equals(hashparam))  Then
          Response.Write("<h4>Security Alert. The digital signature is not valid.</h4>")
                Response.Write("<h4>Generated Hash Val : " + paramsval + "</h4>");
  Response.Write("<h4>Original Hash Val : " + hashparamsval + "</h4>");      End If

       String mdStatus = Request.Form("mdStatus")
       If mdStatus.Equals("1") Or mdStatus.Equals("2") Or mdStatus.Equals("3") Or
    mdStatus.Equals("4") Then
    %>
```

```
    <h5>3D Transaction is Success</h5>
    <br />
    <h3>Payment Response</h3>
    <table border="1">
        <tr>
                <td><b>Parameter Name</b></td>
                <td><b>Parameter Value</b></td>
        </tr>

        <%
        For Each paramname As String In paymentparams
                 Dim paramval As String = Request.Form(paramname)
                 Response.Write("<tr><td>" + paramname + "</td><td>" + paramval + "</td></tr>")
                Next
        %>
    </table>

    <%
       If "Approved".Equals(Request.Form("Response")) Then
    %>
    <h6>Transaction is Success</h6>
    <%
       Else
    %>
    <h6>Transaction is not Success</h6>
    <%
       End If
       Else
    %>
    <h5>3D Transaction is not Success</h5>
    <%
       End If
    End If
    %>
</body>
</html>
```

5.3 JSP Code Sample

5.3.1 Request Sample Codes

5.3.1.1 Hash Version 2

```
<%@page contentType="text/html;charset=ISO-8859-9"%>
```

```
<%@page import="org.apache.commons.codec.binary.Base64" %> <%@page import="java.security.MessageDigest"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-9">
<title>Ver2 Request</title>
</head>
<body>
    <%
            String storeType="3d_pay_hosting";
                  //unEscaped values
            String orgClientId = "xxxxxxxxx";
            String storeType="3d_pay_hosting";
            String orgOid = "";
            String orgAmount = "95.93";
            String orgOkUrl = "http://localhost:8080/SampleCodeJSPTest/GateResponseControl.jsp";
            String orgFailUrl = "http://localhost:8080/SampleCodeJSPTest/GateResponseControl.jsp";
            String orgTransactionType = "Auth";
            String orgInstallment = "";
            String orgRnd = new java.util.Date().toString();

            String orgCurrency = "949";
            // escaped values
            String clientId = orgClientId.replace("\\", "\\\\").replace("|", "\\|");
            String oid = orgOid.replace("\\", "\\\\").replace("|", "\\|");
            String amount = orgAmount.replace("\\", "\\\\").replace("|", "\\|");
            String okUrl = orgOkUrl.replace("\\", "\\\\").replace("|", "\\|");
            String failUrl = orgFailUrl.replace("\\", "\\\\").replace("|", "\\|");
            String transactionType = orgTransactionType.replace("\\", "\\\\").replace("|", "\\|");
                String installment = orgInstallment.replace("\\", "\\\\").replace("|", "\\|");
String rnd = orgRnd.replace("\\", "\\\\").replace("|", "\\|");


        String currency = orgCurrency.replace("\\", "\\\\").replace("|", "\\|");
```

```
        String storeKey = "AB123456".replace("\\", "\\\\").replace("|", "\\|");

                String plainText = clientId + "|" + oid + "|" + amount + "|" + okUrl + "|" + failUrl +
    "|" + transactionType + "|" + installment + "|" + rnd + "||||" + currency + "|" + storeKey;

                MessageDigest messageDigest = MessageDigest.getInstance("SHA-512");
messageDigest.update(plainText.getBytes());
                String hash= new String(Base64.encodeBase64(messageDigest.digest()),"UTF-8");

        String description = "";
        String xid = "";
        String lang="";
        String email="";
        String userid="";
    %>
    <center>
        <form method="post" action="http://localhost:8080/fim/est3dgate">
                <table>
                        <tr>
                                <td>Credit Card Number</td>
                                <td><input type="text" name="pan" size="20" value="" />
                        </tr>
                        <tr>
                                <td>CVV</td>
                                <td><input type="text" name="cv2" size="4" value="" /></td>
                        </tr>
                        <tr>
                                <td>Expiration Date Year</td>
                                <td><input type="text" name="Ecom_Payment_Card_ExpDate_Year"
                                        value="" /></td>
                        </tr>
                        <tr>
                                <td>Expiration Date Month</td>
                                        <td><input type="text"
name="Ecom_Payment_Card_ExpDate_Month"                                          value=""
/></td>
                        </tr>
                        <tr>
                                <td>Choosing Visa / Master Card</td>
                                <td><select name="cardType">
                                        <option value="1">Visa</option>


                                                <option
value="2">MasterCard</option>                                  </select>
                        </tr>
                        <tr>
                                <td align="center" colspan="2"><input type="submit"
    value="Complete Payment" /></td>
```

```
                        </tr>
                </table>

                <input type="hidden" name="clientid" value="<%=orgClientId%>">
                <input type="hidden" name="amount" value="<%=orgAmount%>">
                <input type="hidden" name="oid" value="<%=orgOid%>">
                <input type="hidden" name="okUrl" value="<%=orgOkUrl%>">
                <input type="hidden" name="failUrl" value="<%=orgFailUrl%>">
            <input type="hidden" name="TranType" value="<%=orgTransactionType%>">
                 <input type="hidden" name="Instalment" value="<%=orgInstallment%>">

                <input type="hidden" name="currency" value="<%=orgCurrency%>">
                <input type="hidden" name="rnd" value="<%=orgRnd%>">
                <input type="hidden" name="hash" value="<%=hash%>">
                <input type="hidden" name="storetype" value="<%=storeType%>">
                <input type="hidden" name="lang" value="tr">
                <input type="hidden" name="hashAlgorithm" value="ver2">

        </form>
    </center>
   </body>
</html>
```

5.3.2 Response Code Sample

```
<%@page import="java.util.Enumeration" %>
<%@page import="org.apache.commons.codec.binary.Base64" %>
<%@page import="java.security.MessageDigest"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-9">
<title>Ver2 Response</title>
</head>
<body>
    <h1>Payment Page</h1>
    <h3>Payment Response</h3>
    <table border="1">
    <%
    String originalClientId = "xxxxxxxxx";
          String [] mustParameters = new String[]
{"clientid","oid","Response"};          boolean isValid = true;
```

```
        for(int i=0;i<mustParameters.length;i++){
if(request.getParameter(mustParameters[i]) == null ||
request.getParameter(mustParameters[i]) == "" ){
if(mustParameters[i].equals("oid")){
if(request.getParameter("ReturnOid") == null || request.getParameter("ReturnOid")
== ""
    ){                  isValid = false;                    out.println("<tr><td>Missing
Required Param</td>"+"<td>oid / ReturnOid</td></tr>");

              }
            }else{
      isValid = false;
                        out.println("<tr><td>Missing Required
    Param</td>"+"<td>"+mustParameters[i]+"</td></tr>");
            }
        }
        }
```

```
        if(!request.getParameter("clientid").equals(originalClientId)){
out.println("<h4>Security Alert. Incorrect Client Id.</h4>");
return;
        }
        if(!isValid){
         out.println("<h4>Security Alert. The digital signature is not valid. Required Paramaters
    are missing.</h4>");
           return;
        } else {
    %>
          <tr>
                  <td><b>Parameter Name</b></td>
                  <td><b>Parameter Value</b></td>
          </tr>
          <%
             Enumeration enu = request.getParameterNames();
while(enu.hasMoreElements()){
            String param = (String)enu.nextElement();
                String val = (String)request.getParameter(param);
out.println("<tr><td>"+param+"</td>"+"<td>"+val+"</td></tr>");



          } %>
```

```
                    </table>
                <br>
                <%
 String hashparams = request.getParameter("HASHPARAMS");
 String hashparamsval = request.getParameter("HASHPARAMSVAL");
 String storekey="AB123456";
 String paramsval="";
 String hashval = "";
String hash = "";   int
index1=0,index2=0;
 if(request.getParameter("hashAlgorithm").equals("ver2")){


                                                    String[] parsedParams =
hashparams.split("|");


                                                for(String parsedParam: parsedParams){
                                                 String val = request.getParameter(parsedParam)
== null ? "" :
    request.getParameter(parsedParam);
                                                                                    paramsval +=
val.replace("\\", "\\\\").replace("|",
"\\|") + "|";


                }



                 hashval = paramsval + storekey.replace("\\", "\\\\").replace("|", "\\|");
                 String hashparam = request.getParameter("HASH");


                                                                MessageDigest messageDigest =
MessageDigest.getInstance("SHA-512");
messageDigest.update(hashval.getBytes());
                                                        hash= new
String(Base64.encodeBase64(messageDigest.digest()),"UTF-8");
   } else {
                        do{
                                                    index2 = hashparams.indexOf(":",index1);
                                                     String val =
request.getParameter(hashparams.substring(index1,index2)) == null ? "" :
    request.getParameter(hashparams.substring(index1,index2));
                paramsval += val;
        index1 = index2 + 1;
                   }
                                                while(index1<hashparams.length());


                                            hashval = paramsval + storekey;
```

```
                                                                    String hashparam =
request.getParameter("HASH");

          MessageDigest messageDigest = MessageDigest.getInstance("SHA-512");
messageDigest.update(hashval.getBytes());
                                                              hash= new
String(Base64.encodeBase64(messageDigest.digest()),"UTF-8");
         }     if(!paramsval.equals(hashparamsval)|| !hash.equals(hashparams)) {
         out.println("<h4>Security Alert. The digital signature is not
valid.</h4>");                    out.println("<h4>Generated Hash Val : " +
paramsval + "</h4>");          out.println("<h4>Original Hash Val : " +
hashparamsval + "</h4>");
   }

  String mdStatus = request.getParameter("mdStatus");
   if(mdStatus!=null && (mdStatus.equals("1") || mdStatus.equals("2") || mdStatus.equals("3")||
    mdStatus.equals("4"))) {


   %>
                                <h5>3D Transaction is Success</h5>
                <br />
                        <h3>Payment Response</h3>
                <table border="1">
                <tr>
                                        <td><b>Parameter Name</b></td>
                                        <td><b>Parameter Value</b></td>
                </tr>
                <%
                                    String [] paymentparams = new String[]
    {"AuthCode","Response","HostRefNum","ProcReturnCode","TransId","ErrMsg"};
```

```
          for(int i=0;i< paymentparams.length;i++){
             String paramname = paymentparams [i];
             String paramval = request.getParameter(paramname);
             out.println("<tr><td>"+paramname+"</td><td>"+paramval+"</td></tr>");
            }
        %>
        </table>
        <%
    if("Approved".equalsIgnoreCase(request.getParameter("Response"))){ %>
        <h6>Transaction is Success</h6>
    <%
      }else{  %>
<h6>Transaction is not Success</h6>
```

```
      <% }
    } else { %>
      <h5>3D Transaction is not Success</h5>
      <%}
      }   %>
</body>
</html>
```

5.4 PHP Code Sample

5.4.1 Request Sample Codes

5.4.1.1 Hash Version 2

```
<html>
<head>

<title>3D</title>
<meta http-equiv="Content-Language" content="tr">
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-9">
<meta http-equiv="Pragma" content="no-cache">
<meta http-equiv="Expires" content="now">
</head> <body>
    <?php
        $orgClientId  =   "990000000000001";
        $orgOid =    "ORDER256712jbs\\j6b|";
        $orgAmount = "91.96";
        $orgOkUrl =  "https://www.teststore.com/success.php";
        $orgFailUrl = "https://www.teststore.com/fail.php";
        $orgTransactionType = "Auth";
        $orgInstallment = "";
        $orgRnd =  microtime();

        $orgCurrency = "949";

        $clientId  =  str_replace("|", "\\|", str_replace("\\", "\\\\", $orgClientId));
        $oid =   str_replace("|", "\\|", str_replace("\\", "\\\\", $orgOid));
        $amount = str_replace("|", "\\|", str_replace("\\", "\\\\", $orgAmount));
        $okUrl =  str_replace("|", "\\|", str_replace("\\", "\\\\", $orgOkUrl));
        $failUrl = str_replace("|", "\\|", str_replace("\\", "\\\\", $orgFailUrl));
        $transactionType = str_replace("|", "\\|", str_replace("\\", "\\\\", $orgTransactionType));
        $installment = str_replace("|", "\\|", str_replace("\\", "\\\\", $orgInstallment));
        $rnd = str_replace("|", "\\|", str_replace("\\", "\\\\", microtime()));

        $currency =   str_replace("|", "\\|", str_replace("\\", "\\\\", $orgCurrency));
```

```
        $storeKey =  str_replace("|", "\\|", str_replace("\\", "\\\\", "AB123456\\|"));


        $plainText = $clientId . "|" . $oid . "|" . $amount . "|" . $okUrl . "|" . $failUrl . "|"
.
    $transactionType . "|" . $installment . "|" . $rnd . "||||" . $currency . "|" . $storeKey;

        $hashValue = hash('sha512', $plainText);
        $hash = base64_encode (pack('H*',$hashValue)) ;

        $description = "";
        $xid = "";
        $lang="";
        $email="";
        $userid="";
    ?>

    <center>
        <form method="post" action="https://<host_address>/<3dgate_path>">
                <table>
                        <tr>
                                <td>Credit Card Number</td>
                                <td><input type="text" name="pan" size="20" />
                        </tr>
                        <tr>
                                <td>CVV</td>
                                <td><input type="text" name="cv2" size="4" value="" /></td>
                        </tr>
                        <tr>
                                <td>Expiration Date Year</td>
                                <td><input type="text" name="Ecom_Payment_Card_ExpDate_Year"
                                        value="" /></td>
                        </tr>
                        <tr>
                                <td>Expiration Date Month</td>
                                        <td><input type="text"
name="Ecom_Payment_Card_ExpDate_Month"                                                  value=""
/></td>
                        </tr>
                        <tr>
                                <td>Choosing Visa / Master Card</td>


                                <td><select name="cardType">
                                        <option value="1">Visa</option>
                                        <option value="2">MasterCard</option>
                                </select>
                        </tr>
```

```
                    <tr>
                        <td align="center" colspan="2"><input type="submit"
                        value="Complete Payment" /></td>
                    </tr>
                </table>
                    <input type="hidden" name="clientid" value="<?php echo $orgClientId ?>">
                    <input type="hidden" name="amount" value="<?php echo $orgAmount ?>">
                    <input type="hidden" name="oid" value="<?php echo $orgOid ?>">
                    <input type="hidden" name="okUrl" value="<?php echo $orgOkUrl ?>">
                        <input type="hidden" name="failUrl" value="<?php echo $orgFailUrl
?>">                    <input type="hidden" name="TranType" value="<?php echo
    $orgTransactionType ?>">
                    <input type="hidden" name="Instalment" value="<?php echo $orgInstallment
    ?>">


                      <input type="hidden" name="currency" value="<?php echo $orgCurrency ?>">
                    <input type="hidden" name="rnd" value="<?php echo $orgRnd ?>">
                    <input type="hidden" name="hash" value="<?php echo $hash ?>">
                    <input type="hidden" name="storetype" value="3D_PAY_HOSTING">
                    <input type="hidden" name="hashAlgorithm" value="ver2">
                    <input type="hidden" name="lang" value="tr">
        </form>
    </center>
</body>
</html>
```

5.4.2 Response Code Sample

```
<html>
<head>
<title>3D</title>
  <meta http-equiv="Content-Language" content="tr">
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-9">
  <meta http-equiv="Pragma" content="no-cache">
  <meta http-equiv="Expires" content="now">
</head>
<body>
<h1>3D Payment Page</h1>
<h3>Payment Response</h3>
<table border="1">


<?php
    $originalClientId = "xxxxxx";
      $mustParameters = array("clientid","oid","Response");
    $isValid = true;
   for($i=0;$i<3;$i++)
        {
```

```
        if($_POST[$mustParameters[$i]] == null || $_POST[$mustParameters[$i]] == ""
)
        {
                if($mustParameters[$i] == "oid"){
            if($_POST["ReturnOid"] == null || $_POST["ReturnOid"] == "" ){
               $isValid = false;
               echo "<tr><td>Missing Required Param</td>"+"<td>oid / ReturnOid</td></tr>";
            }
          }else{
               $isValid = false;
                       echo "<tr><td>Missing Required Param</td>"+"<td>"+
    $mustParameters[$i]+"</td></tr>";
          }
         }
       }
       if($_POST["clientid"] != $originalClientId){
          echo "<h4>Security Alert. Incorrect Client Id.</h4>";
return;
       }
if(! $isValid){
          echo "<h4>Security Alert. The digital signature is not valid. Required Paramaters are
    missing.</h4>";
          return;
       } else {
?>
   <tr>
      <td><b>Parameter Name</b></td>
      <td><b>Parameter Value</b></td>
   </tr>
<?php
    $paymentparams =
    array("AuthCode","Response","HostRefNum","ProcReturnCode","TransId","ErrMsg");
    foreach($_POST as $key => $value)
    {
                $check=1;
        for($i=0;$i<6;$i++)
            {               if($key ==
$paymentparams[$i])
                {
                    $check=0;
            break;
                }
        } if($check ==
        1)
```

```
    {                           echo
"<tr><td>".$key."</td><td>".$value."</td></tr>";
        }
    }
?>
</table>
<br>
<br>
<?php
   $hashparams = $_POST["HASHPARAMS"];
   $hashparamsval = $_POST["HASHPARAMSVAL"];
   $hashparam = $_POST["HASH"];
   $storekey="xxxxxx";
   $paramsval="";
   $index1=0;
   $index2=0;
   $escapedStoreKey = "";

    if ($_POST["hashAlgorithm"] == "ver2"){
                $parsedHashParams = explode("|",
$hashparams);               foreach ($parsedHashParams as $parsedHashParam) {
                            $vl =
$_POST[$parsedHashParam];                       if($vl
== null)
                 $vl = "";

                $escapedValue = str_replace("\\", "\\\\", $vl);
                $escapedValue = str_replace("|", "\\|", $escapedValue);
                $paramsval = $paramsval . $escapedValue . "|";
        }
        $escapedStoreKey = str_replace("|", "\\|", str_replace("\\", "\\\\", $storekey));
        $hashval = $paramsval. $escapedStoreKey;
        $hash = base64_encode(pack('H*',hash('sha512', $hashval)));
    } else {
         while($index1 < strlen($hashparams))
         {
                $index2 = strpos($hashparams,":",$index1);
                $vl = $_POST[substr($hashparams,$index1,$index2- $index1)];

                if($vl == null)
        $vl = "";
        $paramsval =
        $paramsval . $vl;
                 $index1 = $index2 + 1;
        }
```

```
$escapedStoreKey = $storeKey;                    $hashval = $paramsval.$escapedStoreKey;
        $hash = base64_encode(pack('H*',sha1($hashval)));
    }
    $hashparamsval = $hashparamsval. "|". $escapedStoreKey;

    if($hashval != $hashparamsval || $hashparam != $hash) {
                echo "<h4>Security Alert. The digital signature is not valid.</h4>"  . " <br
/>\r\n";                    echo "Generated Hash Value : ". $hashval . " <br />\r\n";
        echo "Sent hash value : " . $hashparamsval. " <br />\r\n";               echo
"Generated Hash  : ". $hash . " <br />\r\n";              echo "Sent hash  : " . $hashparam.
" <br />\r\n";
    }

        $mdStatus = $_POST["mdStatus"];
        $ErrMsg = $_POST["ErrMsg"];
    if($mdStatus == 1 || $mdStatus == 2 || $mdStatus == 3 || $mdStatus == 4)
    {
                echo "<h5>3D Transaction is
Success</h5><br/>";  ?>
    <h3>Payment Response</h3>
      <table border="1">
         <tr>
           <td><b>Parameter Name</b></td>
           <td><b>Parameter Value</b></td>
         </tr>
<?php

    for($i=0;$i<6;$i++)
     {
        $param = $paymentparams[$i];
        echo "<tr><td>".$param."</td><td>".$_POST[$param]."</td></tr>";

     }

?>
        </table>
<?php
        $response = $_POST["Response"]; if($response
        == "Approved")
        {


          echo "Payment Process is Successfull";
        } else
        {
        echo "Transaction is not Success. Error = ".$ErrMsg;
```

```
}
} else { echo "<h5>3D Transaction is not
Success</h5>";
}
}
?>
</body>
</html>
```

APPENDIX A: Gateway Parameters

3D 1.0 Mandatory Input Parameters

Those fields are mandatory in request when using 3DS 1.0 authentication standard.
Parameter Name, Description and Format are listed below.

Parameter Name	Description	Format
clientid	Merchant ID	Maximum 15 characters
storetype	Merchant payment model	Possible values: "3d_pay_hosting"
trantype	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	ISO 4217 numeric currency code, 3 digits
oid	Unique identifier of the order	Maximum 64 characters
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

hash	Hash value for client authentication	
hashAlgorithm	Hash version	Use "ver3" and "ver2" are supported for backward compatibility.

3D 2.0 Mandatory Input Parameters

Those fields are mandatory when using EMV 3DS 2.0 authentication standard.
Parameter Name, Lenth/Format, Type and Explanation/Value are listed below.

Parameter Name	**Description**	**Format**
clientid	Merchant number	Alfanumeric, max length: 15
storetype	Merchant payment model, sample values: "3d_pay_hosting"	alfanumeric
Transaction_Type	Transaction type. Sample values: Auth, PreAuth	Alfanumeric
amount	Transaction amount Use "." or "," as decimal separator, do not use grouping character	numeric
currency	ISO code of currency	Numeric, length: 3
oid	Unique identifier of the order	Char, max length: 64
okUrl	The return URL to which Nestpay redirects the customer if transaction is completed successfully. URL: http://www.test.com/ok.ph p	
failUrl	The return URL to which Nestpay redirects the customer if transaction is completed unsuccessfully. http://www.test.com/fail.p hp	
lang	Language of the payment pages hosted by Nestpay "en" for English	
rnd	Random string, will be used for hash comparison	Fix length, 20 char
hash	Hash value for client authentication	

3D 1.0 Optional Input Parameters

Those fields are optional in request when using 3DS 1.0 authentication standard.
Parameter Name, Description and Format are listed below.

Parameter Name	Description	Format
refreshtime	Redirection counter value to okUrl or failUrl in seconds.	Number
encoding	Encoding of the posted data. Default value is "utf-8" if not sent	Maximum 32 characters
description	Description sent to MPI	Maximum 255 characters
comments	Kept as "description" for the transaction	Maximum 255 characters
instalment	Instalment count	Number
GRACEPERIOD	Grace period; postpones the payment of given months	Number (months)
email	Customer's email address	Maximum 64 characters
tel	Customer phone	Maximum 32 characters
BillToCompany	BillTo company name	Maximum 255 characters
BillToName	BillTo name/surname	Maximum 255 characters
BillToStreet1	BillTo address line 1	Maximum 255 characters
BillToStreet2	BillTo address line 2	Maximum 255 characters
BillToCity	BillTo city	Maximum 64 characters
BillToStateProv	BillTo state/province	Maximum 32 characters
BillToPostalCode	BillTo postal code	Maximum 32 characters
BillToCountry	BillTo country code	Maximum 3 characters
ShipToCompany	ShipTo company	Maximum 255 characters
ShipToName	ShipTo name	Maximum 255 characters
ShipToStreet1	ShipTo address line 1	Maximum 255 characters

ShipToStreet2	ShipTo address line 2	Maximum 255 characters
ShipToCity	ShipTo city	Maximum 64 characters
ShipToStateProv	ShipTo state/province	Maximum 32 characters
ShipToPostalCode	ShipTo postal code	Maximum 32 characters
ShipToCountry	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
total1	Subtotal of item #l	Maximum 32 characters
RecurringPaymentNumber	Total number of payments for recurring payment	Number
RecurringFrequencyUnit	Frequency unit for recurring payment	1 char: D=Day,W=Week,M=Month, Y=Year
RecurringFrequency	Frequency of recurring payment	Number
printBillTo	Print BillTo address fields on payment page	"true" or "false". If not sent, billTo address details will not be printed
printShipTo	Print ShipTo address fields on payment page	"true" or "false". If not sent, shipTo address details will not be printed
shopurl	The return URL which NestPay redirects customers when the customer clicks the button "back to order" displayed in HPP.	A valid URL. It is expected from the merchant to send the URL of its website.
CustomerSurcharge	Customer Surcharge amount	Use "." or "," as decimal separator, do not use grouping character
MerchantSurcharge	Merchant Surcharge amount	Use "." or "," as decimal separator, do not use grouping character

CallbackResponse	Enables Auto Capture/Void Functionality	Use "true" or "false"
CallbackURL	URL to send callback for this transaction	A valid URL
DimCriteria1	Merchant specific parameter	Maximum 64 characters
DimCriteria2	Merchant specific parameter	Maximum 64 characters
DimCriteria3	Merchant specific parameter	Maximum 64 characters
DimCriteria4	Merchant specific parameter	Maximum 64 characters
DimCriteria5	Merchant specific parameter	Maximum 64 characters
DimCriteria6	Merchant specific parameter	Maximum 64 characters
DimCriteria7	Merchant specific parameter	Maximum 64 characters
DimCriteria8	Merchant specific parameter	Maximum 64 characters
DimCriteria9	Merchant specific parameter	Maximum 64 characters
DimCriteria10	Merchant specific parameter	Maximum 64 characters
SESSIONTIMEOUT	Session timeout value in seconds for the transaction Minimum and maximum values for Session Timeout are defined by system properties "*gate.statetimeout.min*" and "*gate.statetimeout.max*". If not in range then system property "*gate.statetimeout*" will be used as session timeout value.	Number

3D 2.0 Optional Input Parameters

Those fields are optional when using EMV 3DS 2.0 authentication standard.

Parameter Name, Lenth/Format, Type and Explanation are listed below.

Parameter Name	Description	Format
description		Variable string, max length: 64

ACCOUNTTYPE	01: Not Applicable 02: Credit 03: Debit	string, fix length: 2
CARDHOLDERADDRESSMATCH	Y: Shipping address matches billing address N: Shipping address does not match billing address	Fixed string
CARDHOLDERNAME		variable string, 2-45 characters
Email		variable string, max length: 254
CARDHOLDERHOMEPHONE		variable string, max length: 19(..3-..15)
t e l		variable string, max 19(..3-..15)
CARDHOLDERWORKPHONE		variable string, max length: 19(..3-..15)
TDS2MESSAGECATEGORY	01: Payment 02: Non Payment	fix string, length:2
TDS2PURCHASEDATE	Opt: normal purchases, required if instalment or recurring. Original purchase time in GMT	Format: YYYYMMDDHHMMSS
TDS2TRANSACTIONTYPE	01: Goods/service purchase 03: Check Acceptance 10: Account funding 11: Quasi-cash Transaction 28: Prepaid Activation and Load	fix string, length:2

TDS2REQUESTOR3RIIND	01: Recurring 02:Installment transaction 03:Add card 04: Maintain card information 05:Account verification	fix string, length:2
	05:Account verification. Will be used onlu if deviceCategory=6	
TDS2AUTHENTICATIONINDICATOR	01: Payment 02: Recurring 03: Instalment 04: Add Card 05: Maintain Card 06: Verification as part token CMV token Id	fix string, length:2
TDSCHALLENGEINDICATOR	01: No preference 02: No challenge Requested 03: Challenge Requested 04: Mandate	fix string, length:2
TDS2REQUESTORID		Type: string, max length: 5
TDS2REQUESTORNAME		Type: string, max length: 40
TDS2DSREQUESTORNPAIND	Required if TDS.messageCateg ory=2	Fix string
TDS2REQUESTORURL	Fully qualified URL of 3DS Requestor website Example: http://server.domainname.com	length var string 2048

TDS2WINDOWSIZE	01: 250 * 400 02: 390 * 400 03: 500* 600 04: 600*400 05: Full screen	fix string, length:2
TDS2PAYTOKENID		Type: string, legth: 4
TDSACQUIRERBIN		Type: numeric, length: 1- 11
TDS2ACQUIRERMERCHANTID		Type: string, length: 1- 35
TDS2MERCHANTNAME		Type: string, length: 1- 40
TDS2MCC		type: numeric, length:4
TDS2MERCHANTCOUNTRYCODE		type: numeric, length:3
TDS2MRIINDICATOR	01: Ship to CH's billing address 02: Ship to another verified address on file with merchant 03: Ship to an address different than billing 04: Ship to store 05: Digital Goods 06: Travel or Event tickets 07: Other	fix string, length:2

TDS2MRIDELIVERYTIMEFRAME	01: Electronic Delivery 02: Same Day Shipping 03: Overnight shipping 04: Two day or more shipping	numeric, length: 1-2
TDS2MRIDELIVERYEMAIL		format up to 254, numeric
TDSMRIREORDERINDICATOR	01: First time 02: Reordered	numeric, length 1-2
TDS2MRIPREORDERPURCHASEIND	01: Merchandise available 02: Future availability	numeric, length 1-2
TDS2MRIPREORDERDATE		format 8 character, date format: YYYYMMDD
TDS2MRIGIFTCARDAMOUNT		format max length 15 numeric value in minor units
TDS2MRIGIFTCARDCURRENCY		format 3 numeric characters
TDS2MRIGIFTCARDCOUNT		format 2 numeric characters
TDS2CARDHOLDERACCAGEIND		
TDS2CARDHOLDERACCDATE		Fix YYYYMMDD
TDS2CARDHOLDERACCIND	Length of time that the cardholder has had the account with the 3DS Requestor.	
TDS2CARDHOLDERACCCHANGE	Date that the cardholder opened the account at merchant	Fix YYYYMMDD
TDS2CARDHOLDERACCPASSCHANGEID	Length of time since the cardholder's account information with the 3DS Requestor was last changed.	

TDS2CARDHOLDERACCPASSCHANGE	Date that the cardholder's account with the 3DS Requestor was last changed. Including Billing or Shipping address	Fix YYYYMMDD
TDS2PURCHASECOUNT	Number of purchases with this cardholder account during the previous six months.	var num string ..4
TDS2PROVISIONATTEMPT	Number of Add Card attempts in the last 24 hours	var num string ..3
TDS2TRASACTIONNUMDAY	Number of transactions (successful and abandoned) for this cardholder account with the	var num string ..3
	3DS Requestor across all payment accounts in the previous 24 hours.	
TDS2TRANSACTIONACTYEAR	Number of transactions (successful and abandoned) for this cardholder account with the 3DS Requestor across all payment accounts in the previous year.	var num string ..3
TDS2SHIPADDRESSUSAGEIND	01: this transaction 02: less than 30 days 03: 30-60 days 04: more than 60 days	fix string, length:2
TDS2SHIPADDRESSUSAGE	Date when the shipping address used for this transaction was first used with the 3DS Requestor	Fix YYYYMMDD

TDS2SHIPNAMEIND	01: account name identical to shipping name 02: account name ifferent than shipping name	numeric, length 1-2
TDS2PAYMENTACCIND	01 : No account (guest check-out) 02 : During this transaction 03 : Less than 30 days 04 : 30-60 days 05 : More than 60 days	Length: 2 characters
TDS2PAYMENTACCAGE	Date that the payment account was enrolled in the cardholder's account with the 3DS Requestor	Length: 8 characters JSON Data Type: String Format: YYYYMMDD
TDSSUSPICIOUSACCACTIVITY	01 : No suspicious activity has been observed 02 : Suspicious activity has been observed	Length: 2 characters
BillToCity		Maximum 32 characters
BillToCountry		Max 3 characters
BillToStreet1		Maximum 255 characters
BillToStreet2		Maximum 255 characters
BillToStreet3		Maximum 255 characters
BillToPostalCode		Maximum 32 characters
BillToStateProv		Maximum 32 characters
ShipToCity		Maximum 64 characters
ShipToCountry		Maximum 3 characters
ShipToStreet1		Maximum 255 characters
ShipToStreet2		Maximum 255 characters
ShipToStreet3		Maximum 255 characters
ShipToPostalCode		Maximum 32 characters

ShipToStateProv		Maximum 32 characters
TDSAUTHMETHOD	01: No 3DS Requestor authentication occurred (i.e. cardholder"logged in" as guest) 02: Login to the cardholder account at the 3DS Requestor system using 3DS Requestor's own credentials 03: Login to the cardholder account at the 3DS Requestor system using federated ID 04: Login to the cardholder account at the 3DS Requestor system using FIDO Authenticator	Fix num:2
TDS2AUTHTIMESTAMP	YYYYMMDDHHMM	Fix num:12
TDS2AUTHDATA	Data that documents and supports a specific authentication process.	Var string 2048
TDS2PAIREF	This data element contains a ACS Transaction ID for a prior authenticated transaction	Var string 36
TDS2APAIAUTHMETHOD	01: Frictionless authentication occurred by ACS 02: Cardholder challenge occurred by ACS	Fix num:2
TDS2PAIAUTHTIMESTAMP	YYYYMMDDHHMM (based on UTC)	Fix num:12

TDS2PAIAUTHDATA	Data that documents and supports a specific authentication process.	Var str 2048
TDS2BROADINFO	If not JSON object then in case of 2.1 value is converted to JSON object as {“value” : “value of field” } and passed as is only in case 2.0.	Type: string representing a JSON Object Length: 24096

Note: There are merchant related fields in transaction input parameters, and they will be taken from 3DS Server Database if not provided in the transaction. These fields are shared below:

MERCHANT FIELDS

TDS2REQUESTORID
TDS2REQUESTORNAME
TDS2REQUESTORURL
TDS2ACQUIRERBIN
TDS2ACQUIRERMERCHANTID
TDS2MERCHANTNAME
TDS2MERCHANTCOUNTRYCODE
TDS2MCC

3D 1.0 Transaction Response Parameters

Those fields are in response when using 3DS 1.0 authentication standard. Parameter, Description and Format are listed below.

Parameter	**Description**	**Format**
AuthCode	Transaction Verification/Approval/Authorization code	6 characters
xid	Internet transaction identifier	28 characters
Response	Payment status	Possible values: "Approved", "Error", "Declined"
HostRefNum	Host reference number	12 characters

ProcReturnCode	Transaction status code	Alphanumeric, 2 chars, "00" for authorized transactions, "99" for gateway 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 the 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
HASHPARAMS	Contains the field names used for hash calculation. Field names are appended with ":" character	Possible values "clientid:oid:AuthCode:ProcReturnCode:Response:rnd:" for non-3D transactions, "clientId:oid:AuthCode:ProcReturnCode:Response:mdStatus:cavv:eci:md:rnd:" for 3D transactions
HASHPARAMSVAL	Contains the appended field values for hash calculation. Field values appended with the same order in HASHPARAMS field	Fixed length, 28 characters
HASH	Hash value of HASHPARAMSVAL and merchant password field	Fixed length, 20 characters

3D 2.0 Transaction Response Parameters

Those fields are in response when using EMV 3DS 2.0 authentication standard.
Parameter, Description and Format are listed below.

Parameter	Description	Format
AuthCode	Transaction Verification/Approval/Authorizatio n 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
HASHPARAMS	Contains the field names used for hash calculation. Field names are appended with ":" character	Possible values "clientid:oid:AuthCode:ProcReturnCode:Response:rnd:" for non-3D transactions, "clientId:oid:AuthCode:ProcReturnCode:Response:mdStatus:cavv:eci:md:rnd:" for 3D transactions
HASHPARAMSVAL	Contains the appended field values for hash calculation. Field values appended with the same order in HASHPARAMS field	Fixed length, 28 characters
HASH	Hash value of HASHPARAMSVAL and merchant password field	Fixed length, 20 characters

Example of 3D 2.0 HTTP Form

Example of HTTP form -when using EMV 3DS 2.0 *authentication standard* with parameter set- is described below.

```
<form method="post" action="https://host/fim/est3dgate">
<input type="hidden" name="clientid" value=""/>
<input type="hidden" name="storetype" value=""/>
<input type="hidden" name="hash" value=""/>
<input type="hidden" name=" Transaction_Type" value=""/>
<input type="hidden" name="amount" value=""/>
<input type="hidden" name="currency" value=""/>
<input type="hidden" name="oid" value=""/>
<input type="hidden" name="okUrl" value=""/>
<input type="hidden" name="failUrl" value=""/>
<input type="hidden" name="lang" value=""/>
<input type="hidden" name="rnd" value=""/>
<input type="hidden" name="pan" value=""/>
<input type="hidden" name="Payment_Card_ExpDate_Year" value=""/>
<input type="hidden" name="Payment_Card_ExpDate_Month" value=""/>
<input type="hidden" name="description" value=""/>
<input type="hidden" name="ACCOUNTTYPE" value=""/>
<input type="hidden" name="CARDHOLDERADDRESSMATCH" value=""/>
<input type="hidden" name="CARDHOLDERNAME" value=""/>
<input type="hidden" name="tel" value=""/>
<input type="hidden" name="CARDHOLDERWORKPHONE" value=""/>
<input type="hidden" name="TDS2MESSAGECATEGORY" value=""/>
<input type="hidden" name="TDS2PURCHASEDATE" value=""/>
<input type="hidden" name="TDS2TRANSACTIONTYPE" value=""/>
<input type="hidden" name="TDS2REQUESTOR3RIIND" value=""/>
<input type="hidden" name="TDS2AUTHENTICATIONINDICATOR" value=""/>
<input type="hidden" name="TDSCHALLENGEINDICATOR" value=""/>
<input type="hidden" name="TDS2REQUESTORID" value=""/>
<input type="hidden" name="TDS2REQUESTORNAME" value=""/>
<input type="hidden" name="TDS2DSREQUESTORNPAIND" value=""/>
<input type="hidden" name="TDS2REQUESTORURL" value=""/>
<input type="hidden" name="TDS2WINDOWSIZE" value=""/>
<input type="hidden" name="TDS2PAYTOKENID" value=""/>
```

```
<input type="hidden" name="TDSACQUIRERBIN" value=""/>
<input type="hidden" name="TDS2ACQUIRERMERCHANTID" value=""/>
<input type="hidden" name="TDS2MERCHANTNAME" value=""/>
<input type="hidden" name="TDS2MCC" value=""/>
<input type="hidden" name="TDS2MERCHANTCOUNTRYCODE" value=""/>
<input type="hidden" name="TDS2MRIINDICATOR" value=""/>
<input type="hidden" name="TDS2MRIDELIVERYTIMEFRAME" value=""/>
<input type="hidden" name="TDS2MRIDELIVERYEMAIL" value=""/>
<input type="hidden" name="TDSMRIREORDERINDICATOR" value=""/>
<input type="hidden" name="TDS2MRIPREORDERPURCHASEIND" value=""/>
<input type="hidden" name="TDS2MRIPREORDERDATE" value=""/>
<input type="hidden" name="TDS2MRIGIFTCARDAMOUNT" value=""/>
<input type="hidden" name="TDS2MRIGIFTCARDCURRENCY" value=""/>
<input type="hidden" name="TDS2MRIGIFTCARDCOUNT" value=""/>
<input type="hidden" name="TDS2CARDHOLDERACCAGEIND" value=""/>
<input type="hidden" name="TDS2CARDHOLDERACCDATE" value=""/>
<input type="hidden" name="TDS2CARDHOLDERACCIND" value=""/>
<input type="hidden" name="TDS2CARDHOLDERACCCHANGE" value=""/>
<input type="hidden" name="TDS2CARDHOLDERACCPASSCHANGEID" value=""/>
<input type="hidden" name="TDS2CARDHOLDERACCPASSCHANGE" value=""/>
<input type="hidden" name="TDS2PURCHASECOUNT" value=""/>
<input type="hidden" name="TDS2PROVISIONATTEMPT" value=""/>
<input type="hidden" name="TDS2TRASACTIONNUMDAY" value=""/>
<input type="hidden" name="TDS2TRANSACTIONACTYEAR" value=""/>
<input type="hidden" name="TDS2SHIPADDRESSUSAGEIND" value=""/>
<input type="hidden" name="TDS2SHIPADDRESSUSAGE" value=""/>
<input type="hidden" name="TDS2SHIPNAMEIND" value=""/>
<input type="hidden" name="TDS2PAYMENTACCIND" value=""/>
<input type="hidden" name="TDS2PAYMENTACCAGE" value=""/>
<input type="hidden" name="TDSSUSPICIOUSACCACTIVITY" value=""/>
<input type="hidden" name="BillToCity" value=""/>
<input type="hidden" name="BillToCountry" value=""/>
```

<input type="hidden" name="BillToStreet1" value=""/>

<input type="hidden" name="BillToStreet2" value=""/>

<input type="hidden" name="BillToStreet3" value=""/>

<input type="hidden" name="BillToPostalCode" value=""/>

<input type="hidden" name="BillToStateProv" value=""/>

<input type="hidden" name="ShipToCity" value=""/>

<input type="hidden" name="ShipToCountry" value=""/>

<input type="hidden" name="ShipToStreet1" value=""/>

<input type="hidden" name="ShipToStreet2" value=""/>

<input type="hidden" name="ShipToStreet3" value=""/>

<input type="hidden" name="ShipToPostalCode" value=""/>

<input type="hidden" name="ShipToStateProv" value=""/>

<input type="hidden" name="TDSAUTHMETHOD" value=""/>

<input type="hidden" name="TDS2AUTHTIMESTAMP" value=""/>

<input type="hidden" name="TDS2AUTHDATA" value=""/>

<input type="hidden" name="TDS2PAIREF" value=""/>

<input type="hidden" name="TDS2APAIAUTHMETHOD" value=""/>

<input type="hidden" name="TDS2PAIAUTHTIMESTAMP" value=""/>

<input type="hidden" name="TDS2PAIAUTHDATA" value=""/>

<input type="hidden" name="TDS2BROADINFO" value=""/>

</form>

Example of 3D 2.0 Response Form

Example of Response form -when using EMV 3DS 2.0 authentication standard with parameter set- is described below.

<form name="returnform" action="https://testvpos.asseco-see.com.tr/fim/est3dteststoreutf8" method="post">

<input type="hidden" name="AuthCode" value=""/>

<input type="hidden" name="Response" value=""/>

<input type="hidden" name="HostRefNum" value=""/>

<input type="hidden" name="ProcReturnCode" value=""/>

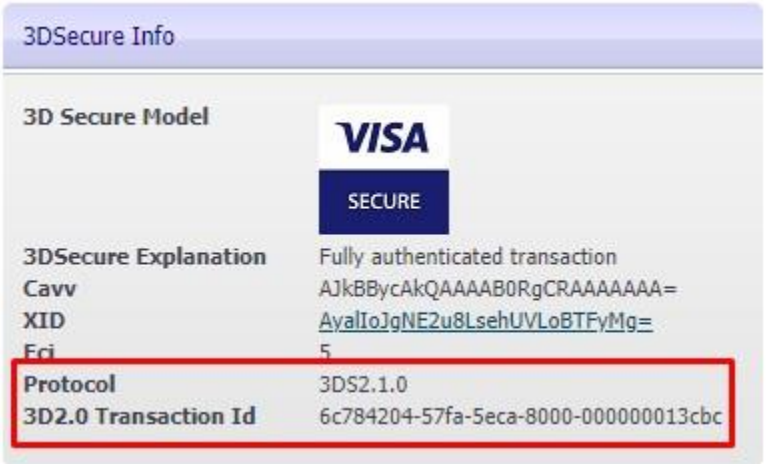
<input type="hidden" name="TransId" value=""/>

<input type="hidden" name="ErrMsg" value=""/>

```
<input type="hidden" name="ClientIp" value=""/>
<input type="hidden" name="ReturnOid" value=""/>
<input type="hidden" name="MaskedPan" value=""/>
<input type="hidden" name="EXTRA.TRXDATE" value=""/>
<input type="hidden" name="rnd" value=""/>
<input type="hidden" name="HASHPARAMS" value=""/>
<input type="hidden" name="HASHPARAMSVAL" value=""/>
<input type="hidden" name="HASH" value=""/>
<input type="hidden" name="mdStatus" value=""/>
<input type="hidden" name="merchantID" value=""/>
<input type="hidden" name="txstatus" value=""/>
<input type="hidden" name="iReqCode" value=""/>
<input type="hidden" name="iReqDetail" value=""/>
<input type="hidden" name="vendorCode" value=""/>
<input type="hidden" name="PAResSyntaxOK" value=""/>
<input type="hidden" name="ParesVerified" value=""/>
<input type="hidden" name="eci" value=""/>
<input type="hidden" name="cavv" value=""/>
<input type="hidden" name="xid" value=""/>
<input type="hidden" name="cavvAlgorthm" value=""/>
<input type="hidden" name="md" value=""/>
<input type="hidden" name="Version" value=""/>
<input type="hidden" name="sID" value=""/>
<input type="hidden" name="MdErrorMsg" value=""/>
<input type="hidden" name="protocol" value=""/>
</form>
```

Nestpay Screens related with 3D authentication

Nestpay shows protocol information, 3D2.0 transaction id, CAVV, XID and ECI on its screens Merchant Center.


3DSecure Info
3D Secure Model
VISA
SECURE
3DSecure Explanation
Fully authenticated transaction
Cavv
AJkBBycAkQAAAAB0RgCRAAAAAAA=
XID
AyalIoJgNE2u8LsehUVLoBTFyMg=
Eci
5
Protocol
3DS2.1.0
3D2.0 Transaction Id
6c784204-57fa-5eca-8000-000000013cbc