POSNET ThreeD Secure

XML Service Integration

Introduction

This document describes how to integrate into the POSNET TDS (3D Secure) system. Shared service urls are for the testing environment. The procedures required to move to the production environment are provided at the end of the document. After completing your tests in the test environment, you must send your request to go live to posnet.support@yapikredi.com.tr. In the mail attachment you will send, you need to include distinctive information (`MERCHANT_ID`, `TERMINAL_ID`, `POSNET_ID`, `SOURCE_IP`, `ORDER_NO`, `TRANSACTION_DATE`, etc.) and the date of the transaction.

Merchants that will use the POSNET system are required to provide Static IP addresses to the bank for both testing and live environments.

It is recommended that the merchant make 3D Secure (3 dimensional security) payment integration both in order to reduce its own risk and to ensure the security of customer information. Cancellation, Refund, Points Transactions, Personal - Joker Vadaa Transactions, Delay Interest Transactions are explained in “POSNET XML Services” document.

The steps necessary for the realization of integration are grouped under 4 main headings.

1. **Encryption of Data:** When the user completes the shopping and arrives at the payment stage, customer information, shopping information and credit card information are sent to YKB services and their data is encrypted. Information such as encrypted posnetData, posnetData2 and digest are contained in response.
2. **User Authentication (3D Secure):** Encrypted data is sent to bank services and user authentication is performed. In this step, a record of the transaction is created on the bank side via the Common Payment Page (OOS) and/or ThreeD Secure (TDS) verification page. After registration, the information required for financialization is sent back to the merchant system in encrypted form.
3. **MAC/User Verification Result Inquiry:** The transaction-specific MAC data is generated by the merchant and sent to bank services with additional information. It is checked by the merchant that the data returned in the service response is the same as the data sent in step 1. This ensures that the information is correctly transferred between the web pages. In addition, the result of the OOS/TDS verification transaction is received with this service. (Merchant that receives card information from their screens and do not perform TDS verification should perform MAC verification)
4. **Financialization:** For financialization, bankData returned from step 2 and the MAC data to be generated are sent to the related service. According to the response of the service, the result of the transaction is provided to the user as information.

The fact that the user verification for financialization has resulted in the beginning or the MAC data has been verified is not controlled in bank systems. The bank accepts no responsibility for the risks that may occur due to the failure of the company to carry out these controls.

- Merchants that will use the POSNET system are required to provide Static IP addresses to the bank for both testing and live environments.

General Structure of the Service

Posnet XML service that enables Posnet merchants to make posnet transactions by sending XML documents. The merchants should POST the xml document after encoding the xml with UTF-8 URL Encode to <%XML_SERVICE_URL%> address (testing environment: https://setmpos.ykb.com/PosnetWebService/XML) as the environment variable at “`xmldata`” parameter with Content-Type=application/x-www-form-urlencoded; charset=utf-8. The result is returned to the merchant as an XML document. <%XML_SERVICE_URL%> shall be an environment variable.

Example URL:
https://setmpos.ykb.com/PosnetWebService/XML?xmldata=%3CposnetRequest%3E%0D%0A++%3Cmid%3E...

The following information included in the service integration is communicated to merchants by mail and this information varies between test and live environments. It is recommended that this information should not be embedded in the code, but should be defined and used as environment variable.

Key	*Type*	*Description*	*Sample Data*
MERCHANT_ID	String	10 digit YKB (Yapı Kredi Bank) merchant number	6706598320
TERMINAL_ID	String	8 digits YKB merchant terminal number	67005551
POSNET_ID	String	Up to 16 digits, YKB merchant POSNET number. It is used in 3D Secure encryption transactions.	9644
ENCKEY	String	Encryption key (fixed for test environment)	10,10,10,10,10,10,10,10
OOS_TDS_SERVICE_URL	String	Address of the bank to which the form will be redirected	https://setmpos.ykb.com/3DSWebService/YKBPaymentService
XML_SERVICE_URL	String	Bank integration service address	https://setmpos.ykb.com/PosnetWebService/XML
MERCHANT_INIT_URL	String	Web address of the merchant	Localhost
MERCHANT_RETURN_URL	String	The merchant page address to which the form will be redirected. Max 255 characters	http://localhost:1453/JavaOOS/merchant_transaction_result.jsp
OPEN_A_NEW_WINDOW	Boolean	Parameter that specifies whether the form to be posted will be redirected to a new page or the current page	0

The MERCHANT_ID, TERMINAL_ID, POSNET_ID, ENCKEY information can also be found on the Merchant information page on the Merchant Admin Screens.

NOTES:

- For each service request, following information shall be added to Request Header: X-MERCHANT-ID, X-TERMINAL-ID, X-POSNET-ID, X-CORRELATION-ID. (CorrelationId: Unique value of the transaction to be set by the merchant, and will allow a quick return of Posnet Support team when a problem is reported. Order number (XID) can be set. If more than one service call is created for the same order, it can be separated by the characters (max 24) to be added to the end of the order number)

- In order to prevent the data to be sent to the service to disrupt the xml structure, xml escape characters must be sent after being encoded.
- UTF-8 encoding is supported in bank systems. The request's content must be set to charset = UTF-8, and the request content must be encoded as UTF-8.

1. Encryption of Data

When the end user reaches the payment step, this is the step to encrypt the payment information and card information. Payment information consists of amount, currency, number of installments, transaction type. Card information consists of name and surname, card number, expiration date and security code. If the merchant does not receive the card information from the user on its screens, it does not send it to the encryption service. In this case, the bank will request the card information from the user via the common payment page.

For encypting the data, the XML structure is created (oosRequestData) and encoded with UTF-8 URL Encode and "`xmldata=`" string is added to the front. The string that starts with `xmldata=%3CposnetRequest%3E%0D%0A++%3Cmid%3E` is posted to < %XML_SERVICE_URL%> with Content-Type=`application/x-www-form-urlencoded; charset=utf-8`

Example URL:
https://setmpos.ykb.com/PosnetWebService/XML?xmldata=%3CposnetRequest%3E%0D%0A++%3Cmid%3E...

Request Example

```xml
<?xml version="1.0" encoding="ISO-8859-9"?>
<posnetRequest>
   <mid>6706022701</mid>
   <tid>67002706</tid>
   <oosRequestData>
      <posnetid>142</posnetid>
      <XID>YKB_0000080603143050</XID>
      <amount>5696</amount>
      <currencyCode>TL</currencyCode>
      <installment>00</installment>
      <tranType>Sale</tranType>
      <cardHolderName>ĞğÜüİıŞşÖöÇç</cardHolderName>
      <ccno>5400637500005263</ccno>
      <expDate>0607</expDate>
      <cvc>111</cvc>
   </oosRequestData>
</posnetRequest>
```

posnetRequest - oosRequestData	
It forms the service input fields that will be used to encrypt the data. posnetData, posnetData2 and digest information will be reached in response.	
posnetRequest	
mid	YKB Merchant Number <%MERCHANT_ID%>
tid	YKB Merchant Terminal Number <%TERMINAL_ID%>
oosRequestData	
posnetid	YKB Merchant POSNET Number <%POSNET_ID%>
XID	Unique shopping order number - 20 alphanumeric characters. The merchant creates it.
amount	Shopping amount - in Kurus Ex: 12.34 TL should be set as 1234.

currencyCode	Currency - “TL, US, EU”
installment	Number of installments “00” should be used for Cash Transaction. "02" should be used for a transaction in installments.
tranType	Transaction Type Sale → Sales Auth → Provision WP → World Points Usage SaleWP → Sales and World Points Usage Vft → Sales with Delay Interest
cardHolderName	Customer's Name and Surname
ccno	Credit Card Number
expDate	Credit card expiry date - In the following format: YY MM
cvc	Credit card security number - CVV2

If it is desired that the bank via the common payment page; cardHolderName, ccno, expDate receives the credit card information, CVC fields are not included in the XML or left blank.

Response Example

```
<?xml version='1.0'?>
<posnetResponse>
    <approved>1</approved>
    <respCode></respCode>
    <respText></respText>
    <oosRequestDataResponse>
        <data1>AEFE78BFC852867FF57078B723E284D1BD52EED8264C6CBD110A1A9EA5EAA7533D1A8
2EFD614032D686C507738FDCDD2EDD00B22DEFEFE0795DC4674C16C02EBBFEC9DF0F495D5E23BE487A79
8BF8293C7C1D517D9600C96CBFD8816C9D8F8257442906CB9B10D8F1AABFBBD24AA6FB0E5533CDE67B0D
9EA5ED621B91BF6991D5362182302B781241B56E47BAE1E86BC3D5AE7606212126A4E97AFC2</data1>

        <data2>69D04861340091B7014B15158CA3C83413031B406F08B3792A0114C9958E6F0F21696
6C5EE32EAEEC7158BFF59DFCB77E20CD625</data2>
        <sign>9998F61E1D0C0FB6EC5203A748124F30</sign>
    </oosRequestDataResponse>
</posnetResponse>
```

posnetResponse - oosRequestDataResponse	
Encrypted data must be saved for later use.	
posnetResponse	
approved	Transaction result. 0: Unsuccessful 1: Successful
respCode	Error code It must be considered when the transaction is unsuccessful. Error Codes section provides explanations.
respText	Error message.
oosRequestDataResponse	
data1	Includes payment information. In the following steps, will be used as **posnetData** variable.
data2	If the card information is in the request, this field is created. In the following steps, will be used as **posnetData** variable.
sign	Service signature. In the following steps, will be used as **digest** variable.

Response Example (Incorrect)

```xml
<?xml version='1.0'?>
<posnetResponse>
    <approved>0</approved>
    <respCode>0002</respCode>
    <respText>XNIException: :::::1:261:cvc-datatype-valid.1.2.1: '569a' değeri 'integer' için geçerli bir değer değil.</respText>
</posnetResponse>
```

2. User Authentication (3D Secure)

It includes the flow through which the user is verified to the card screens by being directed to the bank screens. When the user reaches the payment stage, the merchant system will encrypt the information as described in step 1 and insert the hidden form into the html form with the other information.

Directing the user from the merchant system to OOS/TDS bank pages

```html
<input name="mid" type="hidden" id="mid" value="%=MERCHANT_ID%">
<input name="posnetID" type="hidden" id="PosnetID" value="%POSNET_ID%">
<input name="posnetData" type="hidden" id="posnetData" value="%=DATA1%">
<input name="posnetData2" type="hidden" id="posnetData2" value="%=DATA2%">
<input name="digest" type="hidden" id="sign" value="%=SIGN%">
<input name="vftCode" type="hidden" id="vftCode" value="%=VFT_CODE%">
<input name="useJokerVadaa" type="hidden" id="useJokerVadaa"> <!-- Opsiyonel -->
<input name="merchantReturnURL" type="hidden" id=" merchantReturnURL" value="%=MERCHANT_RETURN_URL%">
<input name="lang" type="hidden" id="lang" value="tr">
<input name="url" type="hidden" id="url" value="">
<input name="openANewWindow" type="hidden" id="openANewWindow" value="%=OPEN_A_NEW_WINDOW%">
```

hiddenFields	
Some of them consist of the environment variables mentioned in the introduction and some of the data obtained from the first step service response. Form variable ids are case sensitive.	
parameters	
mid	YKB Merchant Number <%MERCHANT_ID%>
posnetID	YKB Merchant POSNET Number <%POSNET_ID%>
posnetData	Data block that contains information about shopping (obtained using oosResponseData XML)
posnetData2	Data block that contains credit card information. (obtained using **oosresponsedat** XML.) When this field is left blank, the common payment page will be automatically opened for the user. The user is expected to enter the card information on this screen.
digest	Signature of the FORM to be sent (Obtained using **oosresponsedat** XML.)
vftCode	Determines the campaign code to be used for **Transactions with Delay Interest.** The campaign code defined for the Merchant can be found on the Merchant Information page after logging into Merchant Administrator Screens.
useJokerVadaa	For Member Merchants who will only use the TDS system, it is used to activate the query and use of the Joker Vadaa (additional installment and postponing campaigns specific to member merchants) prior to 3D-Secure verification. It is optional. **This field should not be included in the form if it is not used**.

merchantReturnURL	The address of the page to be redirected to the merchant's site after receiving the card information from the OOS/TDS system or completing the 3D-Secure transaction. If this parameter is not used, it is attempted to redirect to the address registered in the Merchant information page in Posnet Merchant Screens. Max 255 characters <%MERCHANT_RETURN_URL%>
lang	Used to determine the language of the pages in the Posnet system. tr: Turkish en: English
url	Address of redirected page (URL - for information) It is set by Java Script function provided by YKB (in **posnet.js**). It is sufficient to keep it in the form.
openANewWindow	Parameter that specifies whether the form to be posted will be redirected to a new page or the current page The Java Script function supplied by YKB sets it. <%OPEN_A_NEW_WINDOW%>

There are two ways to redirect (POST) the form on the prepared page.

1. Opening a new window and POSTING this form to a new window that opens. Pop-up blocker problems may be encountered and the browser cannot be redirected to the return page of the merchant through the current window due to cross-domain controls in the browsers.
2. Posting the form created in the current window directly to YKB. (**Recommended**)

A JavaScript function written by YKB for the proper routing (**SubmitForm**) can be used. For this, javascript code should be included from https://www.posnet.ykb.com/3DSWebService/scriptler/posnet.js link to the merchant system and must be referenced. Direct use of the link hosted on the bank page is not recommended.

```
<script language="JavaScript" src="https://isyeriadresi/posnet.js"></script>
```

In the corresponding JavaScript function (**SubmitForm**) the following operations are performed before forwarding;

- Setting the "url" parameter as the address of the relevant page,
- If a new window will open, set “openANewWindow” parameter to “0” or “1”,
- If the new window is to be opened, set the value "window.name" to redirect to the main page. Opening window sizes and properties by setting to appropriate values.

```
function submitFormEx(Form, OpenNewWindowFlag, WindowName) {
        submitForm(Form, OpenNewWindowFlag, WindowName)
        Form.submit();
}
```

After the Javascript code is added, the ACTION value of the FORM to be sent during redirection should be changed with environment variable as follows.

```
<form name="formName" method="post" action="%=OOS_TDS_SERVICE_URL%" target="YKBWindow">
```

The environment variable is set to the onclick event value of the form submit button.

```html
<input type="submit" name="Submit" value="Ödeme Yap" onclick="submitFormEx(formName, %=OPEN_A_NEW_WINDOW%, 'YKBWindow')">
```

The html form containing the encrypted data is created in the merchant system and submitted to the user bank screens. If the parameters in the form are missing, format distortion or signature information is incorrect; a warning screen is displayed on the redirected page.

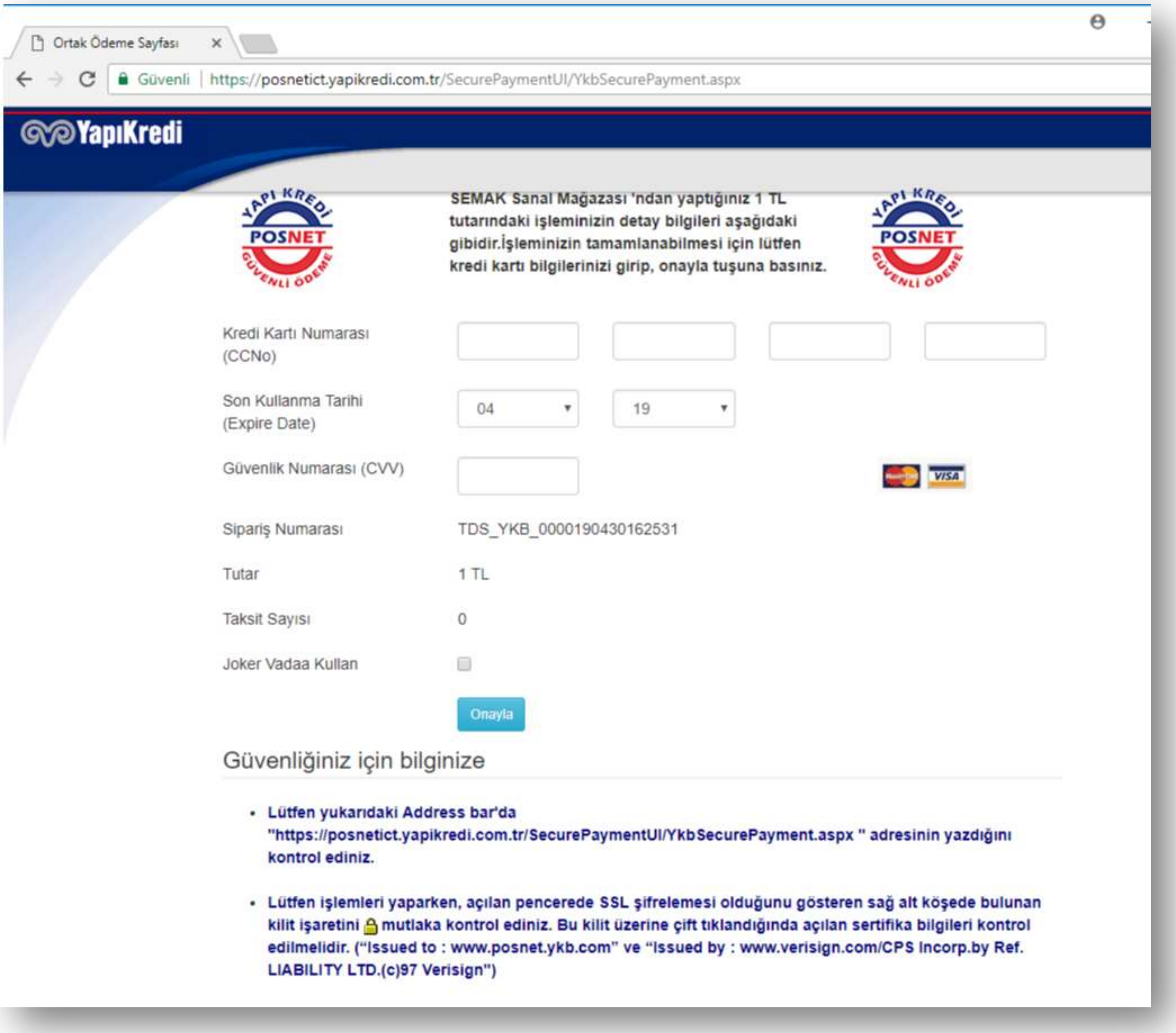
When all fields and scripts are added, the validation form page will consist of an html as follows.

```html
<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta charset="utf-8" />
    <title></title>
    <script type="text/javascript" src="https://www.posnet.ykb.com/3DSWebService/scriptler/posnet.js"></script>
    <script type="text/javascript">
        function submitFormEx(Form, OpenNewWindowFlag, WindowName) {
            submitForm(Form, OpenNewWindowFlag, WindowName)
            Form.submit();
        }
    </script>
</head>
<body>
    <form name="formName" method="post" action="https://setmpos.ykb.com/3DSWebService/YKBPaymentService" target="YKBWindow">
        <input name="mid" type="hidden" id="mid" value="6706598320" />
        <input name="posnetID" type="hidden" id="PosnetID" value="9644" />
        <input name="posnetData" type="hidden" id="posnetData" value="7E2EAA9FCA48B8499C65AB3B820148E7A31F234B439A01C9ECDE8D42101A0F104F985DB3C2D2DA8EA7E7A468030179E17B0632E13E3CE3D7C5096B7593BEE739BD07A0CDE5B46D05FB61FCEB4961F86DCB47B71E567D1E734C3307D6DB31C324151803F1D24D3259B4C28348566886DB82DC6DE2AEA0506FD38E0015403C1A3D52EE8E0CDA8B0043CAAAFE1A93A1B2CDCAD1B12BC7CA1E8A3CDA84EF" />
        <input name="posnetData2" type="hidden" id="posnetData2" value="7585932834B151D962D9CCEE5B5775FCDBDC84E5365F4248E79A453601934B855072D1E36535A8F40BF4F9D478B589AC46ECA928" />
        <input name="digest" type="hidden" id="sign" value="A531D6C260A4573F3753535ED50BE408" />
        <input name="vftCode" type="hidden" id="vftCode" value="" />
        <input name="useJokerVadaa" type="hidden" id="useJokerVadaa" value="1" /> <!-- Opsiyonel -->
        <input name="merchantReturnURL" type="hidden" id=" merchantReturnURL" value="http://localhost:8081/3DSResultPage" />
        <input name="lang" type="hidden" id="lang" value="tr" />
        <input name="url" type="hidden" id="url" value="http://localhost:8080/Payment.html" />
        <input name="openANewWindow" type="hidden" id="openANewWindow" value="0" />

        <input type="submit" name="Submit" value="Doğrulama Yap" onclick="submitFormEx(formName, 0, 'YKBWindow')" />
    </form>
</body>
</html>
```

If there is no encrypted card information (posnetData2) in the submitted form, the card information is taken from the user via the bank common payment page.


Ortak Ödeme Sayfası
Güvenli
https://posnetict.yapikredi.com.tr/SecurePaymentUI/YkbSecurePayment.aspx
YapıKredi
YAPI KREDİ POSNET GÜVENLİ ÖDEME

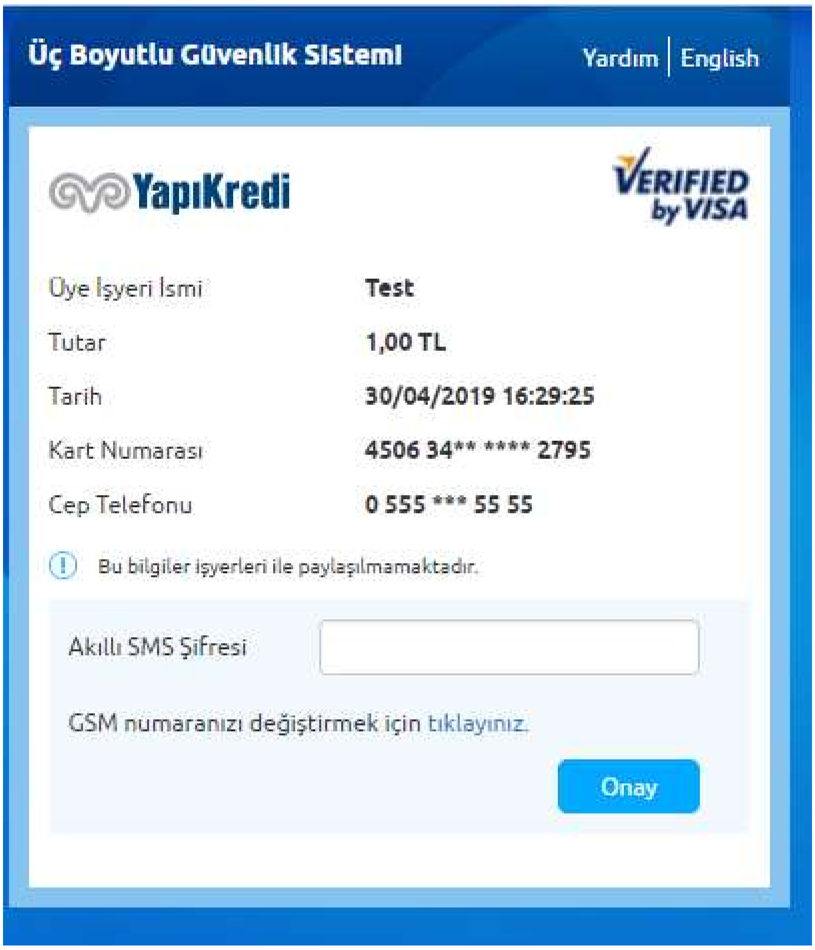
SEMAK Sanal Mağazası 'ndan yaptığınız 1 TL tutarındaki işleminizin detay bilgileri aşağıdaki gibidir.İşleminizin tamamlanabilmesi için lütfen kredi kartı bilgilerinizi girip, onayla tuşuna basınız.

Kredi Kartı Numarası (CCNo)	
Son Kullanma Tarihi (Expire Date)	04 ▾ 19 ▾
Güvenlik Numarası (CVV)	
Sipariş Numarası	TDS_YKB_0000190430162531
Tutar	1 TL
Taksit Sayısı	0
Joker Vadaa Kullan	□

Güvenliğiniz için bilginize

- Lütfen yukarıdaki Address bar'da "https://posnetict.yapikredi.com.tr/SecurePaymentUI/YkbSecurePayment.aspx " adresinin yazdığını kontrol ediniz.
- Lütfen işlemleri yaparken, açılan pencerede SSL şifrelemesi olduğunu gösteren sağ alt köşede bulunan kilit işaretini 🔒 mutlaka kontrol ediniz. Bu kilit üzerine çift tıklandığında açılan sertifika bilgileri kontrol edilmelidir. ("Issued to : www.posnet.ykb.com" ve "Issued by : www.verisign.com/CPS Incorp.by Ref. LIABILITY LTD.(c)97 Verisign")

When the user enters the card information and presses the Confirm button, the cardholder is directed to the bank screen for ThreeD Secure verification. Entering the SMS code sent by the bank to the screen by the user makes verification.



Üç Boyutlu Güvenlik Sistemi

Yardım | English

YapıKredi

VERIFIED by VISA

Üye İşyeri İsmi	**Test**
Tutar	**1,00 TL**
Tarih	**30/04/2019 16:29:25**
Kart Numarası	**4506 34** **** 2795**
Cep Telefonu	**0 555 *** 55 55**

Bu bilgiler işyerleri ile paylaşılmamaktadır.

Akıllı SMS Şifresi

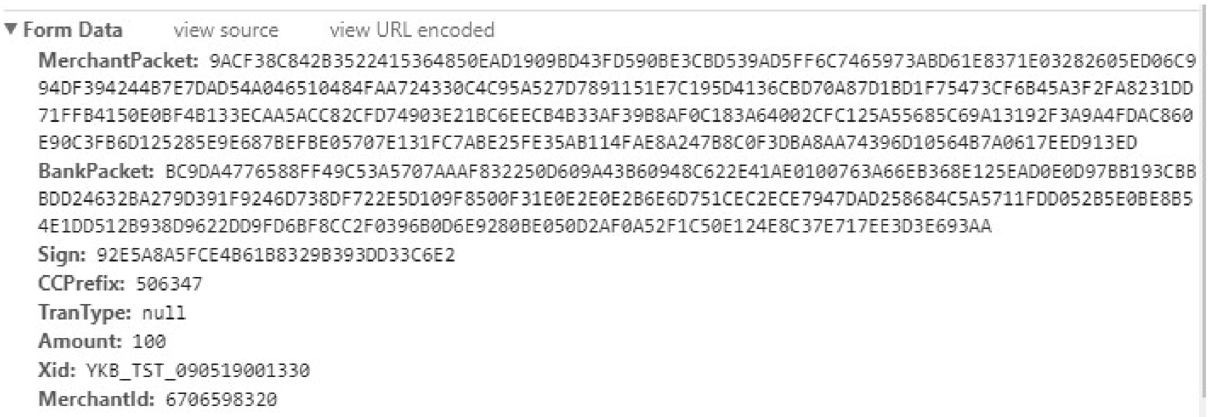
GSM numaranızı değiştirmek için tıklayınız.

Onay

At the end of the 3D-Secure verification process, the user is redirected back to the merchant's systems. During this routing, additional information is sent in the HTML form so that the merchant can obtain the verification result and financialize the relevant transaction. At this stage, the transaction has not been financialized yet; only user/card verification has been performed.

Directing the user from the bank pages to the merchant page

The following data will appear in the routed form. MerchantPackage, BankPackage and Sign data must be read from this data and kept for financialization. Other information is shared for information purposes in order to ensure the integration and transaction security of the merchant and the other information consist of the data that the merchant system reported in the previous steps.



```
▼ Form Data     view source     view URL encoded
    MerchantPacket: 9ACF38C842B3522415364850EAD1909BD43FD590BE3CBD539AD5FF6C7465973ABD61E8371E03282605ED06C9
    94DF394244B7E7DAD54A046510484FAA724330C4C95A527D7891151E7C195D4136CBD70A87D1BD1F75473CF6B45A3F2FA8231DD
    71FFB4150E0BF4B133ECAA5ACC82CFD74903E21BC6EECB4B33AF39B8AF0C183A64002CFC125A55685C69A13192F3A9A4FDAC860
    E90C3FB6D125285E9E687BEFBE05707E131FC7ABE25FE35AB114FAE8A247B8C0F3DBA8AA74396D10564B7A0617EED913ED
    BankPacket: BC9DA4776588FF49C53A5707AAAF832250D609A43B60948C622E41AE0100763A66EB368E125EAD0E0D97BB193CBB
    BDD24632BA279D391F9246D738DF722E5D109F8500F31E0E2E0E2B6E6D751CEC2ECE7947DAD258684C5A5711FDD052B5E0BE8B5
    4E1DD512B938D9622DD9FD6BF8CC2F0396B0D6E9280BE050D2AF0A52F1C50E124E8C37E717EE3D3E693AA
    Sign: 92E5A8A5FCE4B61B8329B393DD33C6E2
    CCPrefix: 506347
    TranType: null
    Amount: 100
    Xid: YKB_TST_090519001330
    MerchantId: 6706598320
```

Information returned in the HTML form;	
Some of them consist of the environment variables mentioned in the introduction and some of the data obtained from the first step service.	
parameters	
MerchantPacket	Transaction details data (merchantData)
BankPacket	Data to be used to financialize the transaction (bankData)
Sign	Data validation information
CCPrefix	The first 6 digits of the credit card used for transaction. Shared for information purposes.
TranType	Transaction type Shared for information purposes.
Amount	Transaction amount. Shared for information purposes.
Xid	Unique shopping order number. Shared for information purposes.
MerchantId	YKB Merchant number. Shared for information purposes.

3. MAC/User Verification Result Inquiry

Creating MAC Data

MAC Data is created in order to compare the amount to be charged on the bank side of the user and the order amount on the merchant side. When creating MAC Data, individual transaction information and environment variables are used on the merchant side. The SHA256 encryption algorithm is given a string converted to UTF-8 byte array and the result is converted to Base64String to complete HASH. For HASH, the 'EncryptionKey' (encKey) and terminalId values are combined with ';' string and firstHash was created after Hashing. Then the xid, amount, currency, merchantNo and firstHash values are combined as string by putting ';' character in between and the HASH operation is completed. Hence, MAC data is obtained.

encKey: 10,10,10,10,10,10,10,10

terminalID: 67005551

when they are used, we need to observe that first data is obtained as follows
c1PPl+2UcdixyhgLYnf4VfJyFGaNQNOwE0uMkci7Uag=.

xid: YKB_TST_190620093100_024

amount: 175

currency: TL

merchantNo: 6706598320

firstHash: c1PPl+2UcdixyhgLYnf4VfJyFGaNQNOwE0uMkci7Uag=

are used, then we observe that MAC data is obtained as follows:
J/7/Xprj7F/KDf98luVfIGyUPRQzUCqGwpmvz3KT7oQ=

JAVA code example

```java
private String HASH(String originalString) throws Exception {
 MessageDigest digest = MessageDigest.getInstance("SHA-256");
 byte[] bytes = digest.digest(originalString.getBytes("UTF-8"));
 return DatatypeConverter.printBase64Binary(bytes);
}

String firstHash = HASH(encKey + ';' + terminalID);
String MAC = HASH(xid + ';' + amount + ';' + currency + ';' + merchantNo + ';' + fir
stHash);
```

C# code example

```
private string HASH(string originalString) {
 using(SHA256 sha256Hash = SHA256.Create()) {
  byte[] bytes = sha256Hash.ComputeHash(Encoding.UTF8.GetBytes(originalString));
  return Convert.ToBase64String(bytes);
 }
}

string firstHash = HASH(encKey + ';' + terminalID);
string MAC = HASH(xid + ';' + amount + ';' + currency + ';' + merchantNo
+ ';' + firstHash);
```

PHP code example

```
Function hashString($originalString){
    return base64_encode(hash('sha256',$originalString,true));

}

$firstHash = hashString($encKey . ";" . $terminalID);
$MAC = hashString($xid . ";" . $amount . ";" . $currency . ";" . $merchantNo  . ";
" . $firstHash);
```

User Authentication Transaction Inquiry

On TDS system, after entering the user verification information, the registration of the user is created on the bank side and directed to the merchant side with additional information in the HTML form. This information is encrypted and they must be deciphered by using **oosResolveMerchantData** service. Before sending the request, the created XML structure is encoded with UTF-8 URL Encode and "`xmldata=`" string is added in front. The string that begins with `xmldata=%3CposnetRequest%3E%0D%0A++%3Cmid%3E`is POSTED to %XML_SERVICE_URL%> with Content-Type=`application/x-www-form-urlencoded; charset=utf-8`

Request Example

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<posnetRequest>
    <mid>6706022701</mid>
    <tid>67002706</tid>
    <oosResolveMerchantData>
        <bankData>87F491ACD24EAE64B519980F0B1BC7547BE4A7C5C614DC3A8CA3FC41B180EE7765
851B081AAE61221956C0C68B0AD69307B4386C7FCE451C272264251BD72BFCBA0A96A197C38C6CD39DD4
42BC179FF098824AFA15B1BB320AD15DA2FB588ECC81B11A26D13764A57B57B49C4CA1BD5D46FA7E60EE
D480C944AE0817</bankData>
        <merchantData>F57E38055C280283044612E7338A314758CE0BB13FE9CFF2D1ACD415A979C1
C65AD1FA664E561809F63262552496B491378DE688980EDFEF32785CB8090E0F3F618D560B4C2C089C7B
9FBA8F91F1F4231D6725ECF8D94B18B0AA9EA206083D94BA1315DCC950E7E5BED2B3B5A1571C3E761E23
64E590CC6BB95BF4F1165208FA55CE99BDE6C7ACDEFB5A2A6F16B6C3838B9876F00EDF1E7261B626532E
E81C40C9DE94588ED36FC4D2E639FA89152D1590A0031416BA8A31A1300EE37E31BD54B6ADA2FF7D4D58
EA0A4A1CC7</merchantData>
        <sign>6C47DAAE1FC76EF98787548B6EBA3B5E</sign>
        <mac>DF2323A3BMC782QOP42RT</mac>
    </oosResolveMerchantData>
</posnetRequest>
```

posnetRequest - oosResolveMerchantData

It is used to query the user verification result and to verify the accuracy of the data.	
posnetRequest	
mid	YKB Merchant Number <%MERCHANT_ID%>
tid	YKB Merchant Terminal Number <%TERMINAL_ID%>
oosResolveMerchantData	
bankData	After the user verification process in step 2, the **bankpacket** data in the html form sent from the bank screens to the merchant screens.
merchantData	After the user verification process in step 2, the **merchantPacket** data in the html form sent from the bank screens to the merchant screens.
sign	After the user verification process in step 2, the **sign** data in the html form sent from the bank screens to the merchant screens.
mac	This is the data that guarantees the accuracy of the transaction between the systems, created by combining the environmental information and the information of the single payment transaction. The URL must be encoded with UTF-8. Please see: Creating MAC Data

Response Example

```
<?xml version='1.0'?>
<posnetResponse>
    <approved>1</approved>
    <respCode></respCode>
    <respText></respText>
    <oosResolveMerchantDataResponse>
        <xid>YKB_0000080603153823</xid>
        <amount>5696</amount>
        <currency>TL</currency>
        <installment>00</installment>
        <point>0</point>
        <pointAmount>0</pointAmount>
        <txStatus>N</txStatus>
        <mdStatus>9</mdStatus>
        <mdErrorMessage>None 3D - Secure Transaction</mdErrorMessage>
        <mac>ED7254A3ABC264QOP67MN</mac>
    </oosResolveMerchantDataResponse>
</posnetResponse>
```

posnetResponse - oosResolveMerchantDataResponse	
merchantData is analyzed by the bank and the MAC Data control confirms that the information is transferred securely. It is absolutely necessary to control that the xid and amount information obtained by decrypting from this package and the xid and amount information used in the sales process by the merchant (step 1 data encryption) are exactly the same.	
posnetResponse	
approved	Transaction result. 0: Unsuccessful 1: Successful
respCode	Error code It must be considered when the transaction is unsuccessful. Error Codes section provides explanations.
respText	Error message.
oosResolveMerchantDataResponse	

xid	Order number
amount	Transaction amount In new Kurus Ex: →12.34 TL should be set as 1234.
currency	Currency TL: Turkish Lira US: American Dollars EU: Euro
installment	Number of installments "00" if Cash Sales
point	World Points information of the credit card used in the transaction Ex: 340
pointAmount	World Points information of the credit card used in the transaction Ex: 170 → 1.70 TL
txStatus	ThreeD Secure Transaction Status
mdStatus	ThreeD Secure Approval Status 0: Card verification failed, do not proceed 1: Verification successful, you can continue with the transaction 2: Card holder or bank is not registered in the system 3: The bank of the card is not registered in the system 4: Verification attempt, cardholder has chosen to register with the system later 5: Unable to verify 6: 3-D Secure error 7: System error 8: Unknown card no 9: Member Merchant not registered to 3D-Secure system (merchant or terminal number is not registered on the back as 3d)
mdErrorMessage	ThreeD Secure Error Message
mac	This is the hashed MAC Data information generated by the bank based on the verification query. By creating a bank response MAC from the merchant, it is necessary to observe that the response is received from the bank and the response has not been altered.

mdStatus indicates the result of the user authentication (3D Secure). A user without user verification can proceed to the financialization step, but in this case the merchant accepts the responsibility. This process is called without 3d verification transaction (NonSecure).

Point and **PointAmount** values can be used in the transactions involving **Sales + Points Usage (Mixed)** transactions. These values return the World Points information available to the cardholder. When making a **Sales + Points Usage** transaction, how many points the cardholder will use (**wpamount**) will be entered on the return pages. Therefore, the Merchant may show these values on their pages, showing the user how many points s/he can use, and allowing the user to use the points accordingly.

Point and **PointAmount** values return “000000000” or an empty value for other operations. For the transaction of **Sales + Points usage**; if the question inquiry is successful, it returns the relevant points, and if it is not successful, it returns “-1”. If the points is shown as "-1", it means that the relevant credit card score information cannot be queried. However, although the point information could not be returned to the merchant, it is possible to continue with **Sales + Points usage** transaction.

Confirmation of Bank Response MAC Data

In order to confirm that **oosResolveMerchantDataResponse** information is received from the bank, MAC data generated by the bank is included as MAC field in **oosResolveMerchantDataResponse** data. It is recommended that the merchant system creates the MAC data itself and compares it with the MAC Data contained in the bank response. If the MAC comparison did not work correctly, it means that the response did not come from the bank. In this case, the transaction should not be continued.

```
String MAC = HASH(mdStatus + ';' + xid + ';' + amount + ';' + currency + ';' + merchantNo + ';' + HASH(EncKey + ';' + terminalID))
```

For mdStatus, as one of the parameters when creating the Mac value, the value on oosResolveMerchantDataResponse object should use xid, amount, currency, merchantNo, EncKey and Terminal ID values, created by the merchant side as first values to be send to encryption service in the first step.

4. Financialization

In order to financialize the transaction, encrypted **bankPacket** data, returned in HTML form when the user is redirected to merchant system at the end of 2nd step is used. For financialization, the XML structure is created (oosTranData) and encoded with UTF-8 URL Encode and “`xmldata=`” string is added to the front. The string that starts with xmldata`=%3CposnetRequest%3E%0D%0A++%3Cmid%3E` is posted to < %XML_SERVICE_URL%> with `Content-Type=application/x-www-form-urlencoded; charset=utf-8`

Request Example

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<posnetRequest>
    <mid>6706022701</mid>
    <tid>67002706</tid>
    <oosTranData>
        <bankData>87F491ACD24EAE64B519980F0B1BC7547BE4A7C5C614DC3A8CA3FC41B180EE7765851B081AAE61221956C0C68B0AD69307B4386C7FCE451C272264251BD72BFCBA0A96A197C38C6CD39DD442BC179FF098824AFA15B1BB320AD15DA2FB588ECC81B11A26D13764A57B57B49C4CA1BD5D46FA7E60EED480C944AE0817</bankData>
        <wpAmount>0</wpAmount>
        <mac>DF2323A3BMC782QOP42RT</mac>
    </oosTranData>
</posnetRequest>
```

posnetRequest - oosTranData	
Controls whether the data is valid and financializes the transaction.	
posnetRequest	
mid	YKB Merchant Number <%MERCHANT_ID%>
tid	YKB Merchant Terminal Number <%TERMINAL_ID%>
oosTranData	
bankPacket	Data used to financialize the transaction (bankData)
wpAmount	In step 1, the data is encrypted when the transaction type is set to SaleWP (Sales + Points Usage-Mixed Transaction). It is in kurus. Ex: 12.34 should be set to 1234 for TL.
mac	This is the data that guarantees the accuracy of the transaction between the

	systems, created by combining the environmental information and the information of the single payment transaction. The URL must be encoded with UTF-8. Please see: Creating MAC Data

Before completing the financialization, the merchant is expected to complete the verification of Bank Response Mac Data, as mentioned in the 3rd step. If the integration between the merchant and the bank is somehow sampled with malware, the way to detect it is to compare encryption using private enc key. If this comparison is not made by the merchant systems, the merchant may suffer financial loss.

To generate the mac data in the oosTranData model, the XID and other information required in the merchant systems must be used.

```
String MAC = HASH(xid + ';' + amount + ';' + currency + ';' + merchantNo + ';' + HASH(encKey + ';' + terminalID));
```

Response Example

```
<?xml version='1.0'?>
<posnetResponse>
    <approved>1</approved>
    <respCode></respCode>
    <respText></respText>
    <mac>DF2323A3BMC782QOP42RT</mac>
    <hostlogkey>0000000002P0806031</hostlogkey>
    <authCode>901477</authCode>
    <instInfo>
        <inst1>00
        <amnt1>000000000000</amnt1>
    </instInfo>
    <pointInfo>
        <point>00000228</point>
        <pointAmount>000000000114</pointAmount>
        <totalPoint>00000000</totalPoint>
        <totalPointAmount>000000000000</totalPointAmount>
    </pointInfo>
</posnetResponse>
```

posnetResponse - instInfo - pointInfo	
The result of the transaction that has financial effect on the user account is included.	
posnetResponse	
approved	Transaction result. 0: Failed, not approved 1: Successful, confirmed 2: Successfully approved before
respCode	Error code It must be considered when the transaction is unsuccessful. Error Codes section provides explanations.
respText	Error message.
mac	This is the hashed MAC Data information generated by the bank based on the financialization request.
hostlogkey	Reference number
authCode	Confirmation code

instInfo - Installment Information	
inst1	Number of installments
amnt1	Number of installment. It is in kurus. Ex: Ex: 12.34 TL should be set as 1234.
pointInfo - Points Information	
point	Points earned
pointAmount	Amount of points earned It is in kurus. Ex: Ex: 12.34 TL should be set as 1234.
totalPoint	Points available to use
totalPointAmount	Total amount of points available It is in kurus. Ex: Ex: 12.34 TL should be set as 1234.
vft - VFT Information	
vftAmount	Delay interest calculated for the transaction It is in kurus. Ex: Ex: 12.34 TL should be set as 1234.
vftDayCount	Number of additional delay days calculated for the transaction

5. Error Codes

The error codes that may be received in case of incorrect parameter entry or connection to the posnet are listed below.

Error Code	*What needs to be done*
100 - OK	Communication with the Posnet server was successfully established. However, this result code does not mean that the transaction is successful. The response from the server needs to be checked to see if the transaction is successful.
101 - CONNECT_ERROR	The connected server ip must be checked.
103 - PACKET_ERROR	This error is returned when Posnet server cannot resolve the packet it receives. Since source ip (ownIP) is used in the analysis process, it should be ensured that this parameter is the same as your IP. The information on the IP Based Errors page can also help you solve the problem.
113 - CONNECT_CONNECT	It should be checked that the hostname parameter is set correctly and has internet connection. By establishing a telnet collection to Hostname parameter (address) in order to control the access to Firewall, etc., the existence of an access problem is controlled. (For example, from the command line: telnet 193.254.228.53 2222). When establishing a telnet connection, make sure that the value entered in the port parameter (2222 unless specified otherwise in the documentation) is also entered in the telnet command (you can only connect to the Posnet server from the correct port, this is also valid for telnet). If you cannot establish a telnet connection to the Posnet server, there is a problem with your internet connection. For example, in your firewall settings, you should ensure that the posnet server uses the correct port. Most firewalls allow only port to connect to http (80) and https (8080). In this case, 2222 (or the connection port specified in the documentation) must be added between the allowed ports. If there is no problem in your internet connection, you should contact the test support group.
115 - CONN_REFUSED	Posnet server refused your connection request. You may have tried a transaction from an IP that is not in the list of IPs, where your company can send transactions to the Posnet system. The information on the IP Based Errors page can also help you solve the problem.
120 - CGI_SERVLET_ERROR	The connection was opened, but the packet could not be sent.
121 - EXCHANGE_TIMEOUT	No response from posnet server. There may be a problem with your Internet connection. If there is no problem with your Internet connection, try again, and if the problem persists, call test support team.
131 - ERROR_CCNO	Card No parameter is incorrect. See parameter descriptions.
132 - ERROR _HOSTLOGKEY	Hostlogkey parameter is incorrect. See parameter descriptions.
133 - ERROR _AUTH	The authorization code parameter is incorrect. See parameter descriptions.
134 - ERROR _HOSTNAME	Hostname parameter is incorrect. See parameter descriptions.
135 - ERROR _PORT	Port parameter is incorrect. See parameter descriptions.
136 - ERROR _OWNIP	Ownip parameter is incorrect. See parameter descriptions.
137 - ERROR _AMOUNT	Amount parameter is incorrect. Before sending the amount, you must make sure that the last two digits are in kurus and that brackets such as

	cents or thousands are not used. For example, you must enter 512 to send 5.12 TL, or 500 to send 5 TL.
138 - ERROR _EXPDATE	The credit card expiration date parameter is incorrect. See parameter descriptions.
139 - ERROR _CVC	The credit card security number (CVC) parameter is incorrect. See parameter descriptions.
140 - ERROR _TAKNUM	The installment parameter is incorrect. The installment parameter must be 2 characters long and should be numeric. Ex: 02. If no installment will be used, 00 or 01 must be entered. Entering the installment parameter 00 or 01 also causes this error in operations that must be in installments (for example; VFT).
142 - ERROR _MIDNO	The merchant number (MID) parameter is incorrect. See parameter descriptions.
143 - ERROR _TIDNO	The terminal number (TID) parameter is incorrect. See parameter descriptions.
144 - ERROR _ORDERID	The order number (ORDERID) parameter is incorrect. It must be 20 characters long and consists of only letters and numbers. Please see the parameter descriptions.
146 - ENCRYPTION ERROR	Encryption error. Send an e-mail to posnet.support@yapikredi.com.tr
147 - CURRENCY CODE ERROR	The currency parameter is incorrect. It is taken when a value other than "TL" or "YT" is entered in the CurrencyCode parameter. The most common cause of this error is to enter "YTL" as a parameter.
156 - ERROR_VFT_CODE	VFT Campaign Code is incorrect. It needs to be 4 characters long.
180 - MULTI AND EXTRA POINTS	It is not possible to specify both the multi points and extra points in the same transaction. You must enter either the multi point parameter 00 or the extra point parameter 000000.
181 - ERROR_TXNSEQNO	The TranSeqNo parameter is incorrect. See parameter descriptions.
184 - ERROR_TRANTYPE	The transaction type parameter is incorrect. See parameter descriptions.
185 - ERROR_BONUS	Points transaction type is incorrect.
186 - ERROR_EXTRAPOINT	Extra points parameter is incorrect. See parameter descriptions.
187 - ERROR_MULTIPLE	Multiple points parameter is incorrect. See parameter descriptions.

If there is no problem in the communication with posnet system and in the parameters (transaction error = 100), the errors that can be received and the actions that should be taken are given below.

Error Code	*Explanation*	*What needs to be done*
0001	BANKANIZI ARAYIN 0001 (CALL YOUR BANK)	The card does not allow this type of transaction or the credit of the card is insufficient. Call the bank that issued the card.
0004	RED-KARTA EL KOY 0004 (REJECT - CONFISCATE THE CARD)	The card is blocked.
0005	RED-ONAYLANMADI (REJECT-DIDN'T APPROVE)	One or more of the card information (Credit card no, expiry date, CVV) may be entered incorrectly or the bank-defined daily limits for World cards may be exceeded. To make sure that the card information is entered correctly, a trial can be performed from the “Online Transactions” page on the Merchant Administrator

		Screens. Receiving this error also means that the card information is sent correctly. Another reason for this error is the *limit of daily transactions, defined by the cardholder bank, to be completed on internet* has been reached. This limit varies according to each bank and it is 3 for YKB credit cards; it means a YKB credit card can be used for up to 3 shopping on internet per day. If this limit is exceeded, the cardholder must call this bank's credit card customer service and reset it. The amount entered cannot be greater than the provision amount in the financialization process and the financialization amount in the refund process.
0007	BANKANIZI ARAYIN 0007 (CALL YOUR BANK)	The card may be blocked/stolen/lost (special case).
0012	RED-GECERSIZ ISLEM (REJECT-INVALID TRANSACTION)	The most common cause of this error is that you try to install with the wrong number of installments. To find out how many installments you can use, you should call 444 0 448. If you are making this transaction with test cards, you may find out the information on posnet.support@yapikredi.com.tr address. Generally, up to 9 installments can be used for normal transactions. Another reason you get error 0012 is that you do something that the card does not allow. For example, you will receive this error if you try to sell with installment on a credit card belonging to another bank. If these steps didn't help you resolve the issue, there may be problems with your bank merchant definitions. By calling our merchant service, you need to give your merchant number and the detail of the transaction that causes this error.
0014	RED-HATALI KART 0014 (REJECT - INCORRECT CARD)	The number does not belong to a credit card/Card number is incorrect.
0015	PROVIZYON BULUNAMADI (NO PROVISION FOUND)	No provision has been placed. Provision may have been canceled. You must place the provision again.
0015	TERMINAL IŞLEM YETKISI YOK (NO TERMINAL TRANSACTION AUTHORITY)	Terminal authorization is not suitable for the transaction.
0015	IŞYERI STATÜSÜ HATALI (MERCHANT STATUS INCORRECT)	Merchant status is not appropriate.
0015	TAKSIT IÇIN YETERSIZ TUTAR (INSUFFICIENT AMOUNT FOR INSTALLMENT)	This error is given if the amount entered for the installment is below the minimum amount.
0030	BANKANIZI ARAYIN 0030 (CALL YOUR BANK)	The reason for this error is the corrupt data sent by the issuer bank to the YKB provision system. The bank that issued the card should be called and indicated that this

		error was received in a virtual pos transaction. In order to find a solution to the problem until the error is resolved, the transaction can be sent to YKB via mail order. To realize a mail order transaction, call our merchant service.
0041	RED-KARTA EL KOY 0041 (REJECT - CONFISCATE THE CARD)	Lost Card - Call (444 0 448).
0043	RED-KARTA EL KOY 0043 (REJECT - CONFISCATE THE CARD)	The cause of the problem is that the credit card used in the transaction is in the **stolen credit card list**, held in YKB provision system. The transaction is rejected before forwarding to the cardholder bank. Credit cards used in virtual POS transactions may be put into a blacklist list by YKB for various reasons. If you believe that the card is incorrectly in the stolen list (the card is a trusted card), you should call the YKB Merchant Operations Service (444 0 448).
0051	RED-YETERSIZ BAKIYE 0051 (REJECT - INSUFFICIENT BALANCE)	The card has insufficient balance. Call the bank that issued the card.
0053	BANKANIZI ARAYIN 0053 (CALL YOUR BANK)	This account isn't found.
0054	RED-ONAYLANMADI 0054 (REJECT- WASN'T APPROVED)	The credit card is expired.
0057	RED-ONAYLANMADI 0057 (REJECT- WASN'T APPROVED)	The transaction cannot be realized with the type of card used (Debit/credit). Example: POSNET cannot process debit cards (debit cards are used to withdraw money from ATMs). In the error message, where ”X" is specified, the type of the card is specified (D:Debit/K: Credit card).
0057	RED-ONAYLANMADI 0057 (REJECT- WASN'T APPROVED)	This error is received when there is a problem with the authorization of the credit card used in the transaction to make transactions from the internet. The cardholder should contact the credit card service of the bank where s/he receives the credit card and indicate that s/he cannot use the credit card in e-commerce.
0058	RED-ONAYLANMADI 0058 (REJECT- WASN'T APPROVED)	The terminal is not authorized for the transaction type.
0062	RED-ONAYLANMADI 0062 (REJECT- WASN'T APPROVED)	Restricted card.
0065	RED-ONAYLANMADI 0065 (REJECT- WASN'T APPROVED)	This error, given when the credit card withdrawal limit is exceeded, should not be returned in virtual pos transactions under normal circumstances. If this error is received, the issuer bank should be called and it should be stated that the error has been received in a virtual pos transaction. In order to find a solution to the problem until the error is resolved, the transaction can be sent to YKB via mail order. To realize a mail order transaction, call our merchant service.
0091	BANKANIZI ARAYIN 0091 (CALL YOUR BANK)	There was a timeout in communication with the issuer bank (no timely response from the bank). Try again; if

		the problem persists, call the issuing bank and indicate that this error was received in a virtual pos transaction.
0100	HOST RECEIVE PROBLEM	This error can sometimes be received when there are instant problems in our bank systems. Try again, if the problem persists, contact posnet.destek@ykb.com . If this error occurs in the test environment, deleting the definitions of the test card used may cause the problem. To eliminate this possibility, you may need to try with several different test cards.
0122	DATABASE DE ISTENILEN KAYIT YOK (REQUIRED RECORD DOESN'T EXIST ON DATABASE)	Error in cancellation. Cancellation can be done up to 1 week after the provision. This error may also be received if the financialization is canceled without completing the financialization. One reason for this error is that you want to financialize or cancel a transaction you have already done with your merchant mid using another mid of your company. The most common way to do this is to programmatically financialize or cancel a transaction made using one mid programmatically using another mid.
0123	ORJINAL ISLEM BULUNAMADI (ORIGINAL TRANSACTION CANNOT BE FOUND)	The transaction to be financialized, refunded or cancelled cannot be found. You are probably trying to financialize/cancel with the wrong YKB ref.no or order no. The transaction you are trying to financialize/cancel may not have been sent to the Posnet system at all. In the cancellation ofVFT transactions, if the cancellation is made with YKB ref. no, the authentication code is also checked. In case of cancellation, the confirmation code should be checked together with YKB ref.no. When no response is received from the Posnet system for a transaction, it is normal to receive this error upon automatic cancellation; this means that the transaction never reaches the Posnet system.
0124	HOST SESSION OPEN PROBLEM	This error is due to the environment of our bank. Occasionally, there are instant interruptions due to the work in our Bank's environments. If this error is received, the transaction should be retried after a while, if the problem persists, posnet.destek@ykb.com should be contacted.
0125	ORDERID VAR HOSTLOGKEY YOK DB ERR (ORDERID EXISTS, HOSTLOGKEY DOESN'T EXIST, DB ERR)	Call YKB.
0126	ORDERID VAR KK SIFRELEME HATASI (ORDERID EXISTS, CREDIT CARD ENCRYPTION ERROR)	Call YKB.
0127	ORDERID DAHA ONCE KULLANILMIS (ORDERID HAS	The order no (orderId) you are using has been previously used. Try again with a new order no.

	BEEN USED BEFORE)	
0129	KREDI KARTI MERCHANT BLACKLIST TE (CREDIT CARD IS IN THE MERCHANT BLACKLIST)	This credit card is included into the merchant blacklist. The card must be removed from the blacklist before the merchant can use it.
0146*	HATALI SIFRELEME : KULLANICI ISMI & SIFRE veya NO GENERATED RECORD (ERROR IN ENCRYPTION: USER NAME & PASSWORD OR NO GENERATED RECORD)	The user name, password or encryption key is entered incorrectly. Please check StubF1Class.setUserName, StubF1Class.setPassword, StubF1Class.setEncKey methods for more information. It is necessary to use "Create Key" on the main menu of Merchant Administrator Screens and new user name, password and key must be generated and retry this transaction with new information.
0147*	HATALI KULLANICI ISMI & SIFRE (ERROR IN USER NAME & PASSWORD)	See explanations of error 146.
0148*	CRYPTO HATASI : MID (ERROR IN CRYPTO: MID)	Your web server's date, time, or Time Zone information may be incorrect. If there is no problem with this information, please contact our Technical Support. Posnet Service, which responds to the information you send, uses date and time to open some encrypted information. If your server's date or time is incorrect, this information cannot be resolved by the service.
0148*	HATALI MID (ERROR IN MID)	The merchant number cannot be found. The merchant no (MID) parameter is incorrect.
0148*	MID,TID,IP HATALI: X.X.X.X (ERROR IN MID, TID, IP)	You are trying to make a connection from a wrong or unauthorized IP when making a connection. Sending a process to the wrong environment (for example, live environment mid and test environment) also causes this problem. For the test environment, you need to send the transaction to **https://setmpos.ykb.com/posnetwebservice/xml** while you need to send the transaction to **https://www.posnet.ykb.com/posnetwebservice/xml** for live environment. If you are sure that you are sending your transaction to the correct environment, you can change your IP definition by sending your request ip to posnet.destek@ykb.com with mid/tid as indicated in the error message XXXX.
0150	PAKET HATALI (ERROR IN PACKET)	Wrong CVC number is used. This error is received if XXX used in the live environment is used in the test environment. In the live environment, your customer must enter the CVC code. In addition, entering a meaningless CVC (such as xxx) other than XXX in the test environment will also cause this error.
0150	INVALID MID TID IP	You are trying to make a transaction from a wrong IP or a wrong mid/tid. The information on the IP Based Errors

		page can also help you solve the problem.
0200	GECERSIZ ISLEM (INVALID TRANSACTION)	Received when you submit an invalid transaction. For example, attempting to refinance a transaction that has already been financialized, or to refund a transaction that is in provision status. This type of invalid transaction is not already allowed on the Merchant Administrator Screens, but this control is performed for the transactions sent in the program (using technology such as ASP).
0205	GECERSIZ TUTAR (INVALID AMOUNT)	This error is received under the following conditions: - When the amount of the transaction exceeds the maximum transaction amount (99.999.99 TL). Up to 99.999.99 TL, a transaction can be made in Posnet system at one time. - While financializing the amount of the transaction exceeds the provision-overrun percentage . - In return transactions, when the transaction amount exceeds the refundable amount.
0211	GROUP CLOSING COMPLETED	This error is received when making financialization or sales cancellation. The transaction you want to cancel is financialized and can no longer be canceled. To return your financialization or refund the sales, you need to make a return transaction.
0217	GEÇERSIZ IŞLEM STATÜSÜ (INVALID TRANSACTION STATUS)	Stolen card. It is necessary to notify YKB about the user name and card number.
0220	IPTAL ISLEMI YAPILMIS (CANCELLATION COMPLETED)	This error is received when you try to cancel again a transaction already cancelled.
0223	ONAYLANMADI (WASN'T APPROVED)	Although the financialization is not completed, financialization is requested to be canceled.
0232	KREDIKARTI IŞLEM SINIRI AŞILDI (CREDIT CARD LIMIT EXCEEDED)	When the maximum number that can be processed with a credit card is exceeded in a certain period defined by the Merchant in the Posnet system, the related error is received. See. Transaction Restriction
0370	ISLEM IPTALI YAPILMIS (TRANSACTION CANCELLED)	The cancellation has already been done.
0400	DB ERROR	Posnet server is having a technical problem. Try again, if the problem repeats, contact the Technical support team.
0411	ISLEM HENUZ FINANSALLASMAMIS (TRANSACTION NOT YET FINANCIALIZED)	This error received when making a refund indicates that the amount specified in the financialization transaction has not yet been collected from the card and reflected in your account. Therefore, you do not need to make a refund transaction; you must cancel the financialization.
0444	BANKANIZI ARAYIN (CALL YOUR BANK)	Call YKB.
0450	IADE ISLEMI YAPILAMIYOR (RETURN TRANSACTION CANNOT	It may be refunded from a screen other than the merchant administrator screen. You may have been

	BE COMPLETED)	refunded the transaction by calling our merchant service. If you did not request such a refund, you should call our merchant service.
0788	FINANSAL ISLEM YAPILMIS (FINANCIAL TRANSACTION COMPLETED)	Financialization is completed. If you want to cancel a provision, financialization must be canceled first.

Steps for Going Live

After completing your tests in the test environment, you must send your request to go live to posnet.support@yapikredi.com.tr. In the mail attachment you will send, you need to include distinctive information (MERCHANT_ID, TERMINAL_ID, POSNET_ID, SOURCE_IP, ORDER_NO, TRANSACTION_DATE, etc.) and the date of the transaction.

For each service request integration, following information shall be added to Request Header: X-MERCHANT-ID, X-TERMINAL-ID, X-POSNET-ID, X-CORRELATION-ID.

1. The MERCHANT_ID, TERMINAL_ID, POSNET_ID information can also be found on the Merchant information page on the Merchant Admin Screens.
2. If the environment variables and XML_SERVICE_URL are used, OOS_TDS_SERVICE_URL is added to the merchant live environment application.
3. Live environment IP information is defined to the system through merchant management screens.

Merchant application configurations are updated so that variables defined as environment variables are used in a live environment.

Key	*Type*	*Description*	*Sample Data*
MERCHANT_ID	String	10 digit YKB (Yapı Kredi Bank) merchant number	6706598320
TERMINAL_ID	String	8 digits YKB merchant terminal number	67005551
POSNET_ID	String	Up to 16 digits, YKB merchant POSNET number. It is used in 3D Secure encryption transactions.	9644
XML_SERVICE_URL	String	Bank integration service address	https://www.posnet.ykb.com/PosnetWebService/XML
OOS_TDS_SERVICE_URL	String	Bank common payment and 3D Secure page address	https://www.posnet.ykb.com/3DSWebService/YKBPaymentService
ENCKEY	String	Encryption Key	<%LIVE_ENCKEY %>
MERCHANT_INIT_URL	String	Web address of the merchant	https://www.example.com
MERCHANT_RETURN_URL	String	The merchant page address to which the form will be redirected. Max 255 characters	https://www.example.com/PaymentResult
OPEN_A_NEW_WINDOW	Boolean	Parameter that specifies whether the form to be posted will be redirected to	0

		a new page or the current page	

If the merchant is making 3D secure payments or using the common payment page provided by Posnet, it means that 3D Secure is active and the customer of the merchant, so the end user, will be directed from the merchant screens to the bank screens and sent back to the merchant screen after passing through the security and verification steps on the bank screens. MAC validation is performed on 3DS payment flows in order to prevent the customer's movement between networks. To create MAC data, it is necessary to follow the Key Generation step from the merchant management screens and set an **ENCYKEY** value for the live environment. It should be noted that this value does not contain Turkish characters and spaces.

History

Date	Version	Explanation	Prepared by
12.05.2019	2.0	A platform independent integration document was created by using the reference of the documentation, prepared on the development environment (.net, java, php, etc.). • Encryption of Data • User Verification • MAC/User Verification Result Inquiry • Financialization • Error codes	Kemal Koray Pekdemir - Virtual Pos and Campaign Application Development
20.06.2019	2.0.1	MAC creation has been added to PHP codes. Format adjustments have been made.	Nazım Sezer - Virtual Pos and Campaign Application Development