

---

# SanalPos Entegrasyon Dökümanı

---

## 3D Model Ödeme

---

Kuveyt Türk Katılım Bankası  
Versiyon 2

---

## İçindekiler

GİRİŞ.....	4
3D Secure .....	4
3D Model.....	4
HIZLI BAŞLANGIÇ KILAVUZU .....	4
Açıklama.....	4
Entegrasyon .....	5
Kullanılan Mesaj yapıları .....	6
Ödeme İşlemi(Onayı) Mesaj yapısı.....	6
Asp.Net.....	6
Ödeme İşlemi Mesaj Örneği.....	7
Asp.Net.....	7
Ödeme İşlemi Onayı Mesaj Örneği .....	7
Asp.Net.....	7
Ödeme İşlem Sonucu Mesaj Yapısı .....	8
Asp.Net.....	8
HashData .....	9
Örnek Kod Parçaları .....	9
Asp.Net.....	9
Java.....	10
Php .....	11
Ödeme İşlemi Yapılması .....	11
Asp.Net.....	11
İşlem Sonucunun Alınması .....	12
Asp.Net.....	12
Ödeme İşleminin Onaylanması .....	12
Asp.Net.....	12
SanalPos 3D Model Ödeme Noktası Adresi (Production) .....	13
SanalPos 3D Model Ödeme Onaylama Adresi (Production) .....	13
SanalPos 3D Model Ödeme Noktası Adresi (Test) .....	13
SanalPos 3D Model Ödeme Onaylama Adresi(Test).....	13
SanalPos 3D Model Ödeme Noktası Adresi (PreProduction).....	13

---

SanalPos 3D Model Ödeme Onaylama Adresi(PreProduction) .....	13
--	----

## GİRİŞ

### 3D Secure

Visa ve MasterCard ın dahil olduğu, internet üzerinden gerçekleştirilen ödeme işlemlerinde daha güvenilir bir ortam sağlayan sistemdir.

3D Secure sistemi kullanıcı, üye işyeri ve banka arasındaki sorumlulukları da düzenlemektedir.

3D Secure sistemi uygulanan ödeme işlemlerinde kullanıcıdan kart bilgileri alındıktan sonra, kullanıcı bankanın 3D Secure ödeme sayfasına yönlendirilmektedir. Burada kullanıcı banka tarafından kendisine gönderilen kısa mesajta yer alan şifreyi girer, böylece kart doğrulanmış ve yapılan işlem kullanıcı tarafından imzalanmış olmaktadır.

Kuveyt Türk Katılım Bankası tarafından sunulan SanalPos servisi 3D Secure ödeme işlemlerini desteklemektedir.

### 3D Model

Üye işyerlerinin 3D Secure sistemini uygulayabilecekleri birçok yöntem bulunmaktadır. **3D Model** bu yöntemlerden birisidir. Bu yöntem 3D Secure sisteminin faydaların tümünü sağlamaktadır. Bu yöntem ayrıca; üye işyerine kart doğrulama, kart doğrulaması sonrasında işlemi tekrar kontrol edebilme ve uygun olması durumunda provizyon işlemine onay verebilme imkanı sağlamaktadır.

3D Model uygulanmış tipik bir ödeme işlemi aşağıdaki temel adımlarla gerçekleşmektedir.

- Kullanıcının karta ait bilgileri girmesi,
- Kart sahibi banka tarafından kartın doğrulanması, kullanıcının işlemi imzalaması,
- Doğrulama sonucunun üye işyeri tarafından kontrol edilmesi,
- Ödemenin gerçekleştirilmesi.

Bu adımların uygulanmasına dair teknik bilgiler daha sonraki bölümlerde yer almaktadır.

## HIZLI BAŞLANGIÇ KILAVUZU

### Açıklama

Bu bölümde üye işyeri ve SanalPos servisi arasında gerçekleştirilecek entegrasyon için bilgiler yer almaktadır. Üye işyerinin 3D Model ödeme uygulayabilmesi için aşağıdaki bilgilere sahip olması gerekmektedir.

Banka tarafından sağlanacak bilgiler aşağıdaki gibidir;

- *Müşteri No(CustomerNumber)*: Üye işyerinin Kuveyt Türk'te yer SanalPos için kullanılabilecek hesaba ait müşteri numarasıdır.

- **Üye İşyeri No(MerchantId):** Üye işyerinin Kuveyt Türk SanalPos servisinde kayıtlı özel numarasıdır.
- **Kullanıcı Adı(Username):** Üye işyerinin SanalPos servisini kullanabileceği sistem kullanıcı adıdır.
- **Ödeme Noktası Adresi:** SanalPos'un sunduğu ödeme yöntemine ait adrestir. Ödeme işlemine ait bilgiler bu adrese gönderilecektir.(Bkz. “SanalPos 3D Model Ödeme Noktası Adresi”)

Entegrasyon için kullanılacak diğer veriler aşağıdaki gibidir;

- **İşlem No(OrderId):** İşlemin SanalPos sisteminde kayıtlı numarasıdır.
- **Sipariş No(MerchantOrderId):** İşlemin üye işyeri tarafında yer alan numarasıdır.
- **Başarılı İşlem Adresi(OkUrl):** 3D Secure işlemlerinde kartın/müşterinin doğrulanması aşamasında, kullanıcı bir başka sayfaya yönlendirilmektedir. Bu aşamada işlemler arasında bağlantı bulunmamaktadır. Bu nedenle işlemlere ait sonuçlar üye işyerinin belirleyeceği bir sayfaya gönderilmektedir. Başarılı 3D Secure işleme ait sonucun gönderileceği adrestir.
- **Başarısız İşlem Adresi(FailUrl):** Başarısız 3D Secure işleme ait sonucun gönderileceği adrestir.
- **Doğrulama Anahtarı(HashData):** Ödeme işlemine ait bilgilerin doğruluğunun kontrol edilmesi için kullanılan veridir. Daha sonra sağlanacak algoritma ile üretilen bu veri, SanalPos tarafında da üretilen ve bu veriler karşılaştırılacaktır. Bu kontrolden geçemeyen ödemeler gerçekleştirilmemektedir.
- **İşlem Güvenliği(TransactionSecurity):** İşlemin yapılacağı güvenlik türü bilgisidir.3D Secure işlemler için 3 değeri girilmelidir.
- **Tutar(Amount):** İşlem tutarı virgülsüz olarak gönderilmelidir. Örneğin 12.34 TL(12 TL 34 Kr) için 1234 gönderilmelidir, yada 12.03 TL için 1203 gönderilmelidir.

## Entegrasyon

SanalPos tarafından sunulan 3D Model ödeme yöntemi için kullanılacak örnek bir entegrasyon işlemi bu kısımda anlatılmaktadır.

Üye işyeri sipariş detayını görüntüler. Kullanıcıdan, 3D Secure ödeme işlemini seçmesi beklenir.

Üye işyeri kullanıcıdan kart bilgilerini alır. Kart, sipariş ve üyeişyerine ait bilgileri içeren mesaj oluşturulur. Mesajda yer alacak HashData alanına dikkat edilmelidir. Bu mesaj yapısı XML formatındadır. Bir sonraki bölümde mesaj yapısı yer ve örnek mesaj yer almaktadır.(Bkz. “Ödeme İşlemi Yapılması”)

Üye işyeri oluşturduğu XML mesajı SanalPos tarafından sunulan 3D Model ödeme noktasına HTTP POST yöntemi ile gönderir. Bu işlem sonucunda HTML yapısında bir cevap yer almaktadır. Üye işyeri bu HTML cevabı tarayıcıya iletmelidir.(Bkz. “Ödeme İşlemi Yapılması”, “SanalPos 3D Model Ödeme Noktası Adresi”)

SanalPos tarafından gerçekleştirilecek işlemler sonrasında işleme ait sonuç XML mesaj şeklinde FailUrl ya da OkUrl e HTTP Post ile iletilecektir. Bir sonraki bölümde mesaj yapısı ve örnek mesaj yer almaktadır.(Bkz. “İşlem Sonucunun Alınması”)

Üye işyeri FailUrl'e veya OkUrl'e gelecek cevaba göre kontrollerini yapar, dilerse kullanıcıya gösterebilir. Eğer işlem sonucu başarılı ise(mesaj OkUrl'e iletildi ise) üye işyeri ödeme işlemini sonlandırmak için SanalPos 3D Model Provizyon adresine ödeme mesajı göndermelidir.Gönderilecek mesajta, bu noktada gelen cevapta yer alan "MD" değeri ile birlikte ilk aşamada gönderilen üye işyeri ve sipariş bilgileri gönderilmelidir. SanalPos 3D Model Provizyon adresine gönderilen mesaj SanalPos servisi tarafından işlenir ve XML cevap verilir. XML cevapta yer alan ResponseCode değeri "00" olduğu durumda ödeme işlemi gerçekleşmiş olmaktadır.(Bkz. "Ödeme İşleminin Onaylanması", "SanalPos 3D Model Ödeme Onaylama Adresi")

## Kullanılan Mesaj yapıları

### Ödeme İşlemi(Onayı) Mesaj yapısı

#### Asp.Net

```
<KuveytTurkVPosMessage>
  <APIVersion></APIVersion>
  <OkUrl></OkUrl>
  <FailUrl></FailUrl>
  <HashData></HashData>
  <MerchantId></MerchantId>
  <CustomerId></CustomerId>
  <UserName></UserName>
  <CustomerIpAddress></CustomerIpAddress>
  <CardNumber></CardNumber>
  <CardExpireDateYear></CardExpireDateYear>
  <CardExpireDateMonth></CardExpireDateMonth>
  <CardCVV2></CardCVV2>
  <CardHolderName></CardHolderName>
  <CardHolderIpAddress></CardHolderIpAddress>
  <CardType></CardType>
  <TransactionType></TransactionType>
  <InstallmentCount></InstallmentCount>
  <Amount></Amount>
  <DisplayAmount></DisplayAmount>
  <Description></Description>
  <CurrencyCode></CurrencyCode>
  <MerchantOrderId></MerchantOrderId>
  <TransactionSecurity></TransactionSecurity>
  <KuveytTurkVPosAdditionalData>
    <AdditionalData>
      <Key></Key>
      <Data></Data>
      <Description></Description>
    </AdditionalData>
  </KuveytTurkVPosAdditionalData>
</KuveytTurkVPosMessage>
```

Not: Doğrulama sonrası gönderilecek ödeme mesajında MD değeri AdditionalData altında aşağıdaki gibi gönderilmelidir.

```
<KuveytTurkVPosAdditionalData>
  <Key>MD</Key>
  <Data>OkUrl e gelen XML de yer alan MD değeri</Data>
  <Description></Description>
</KuveytTurkVPosAdditionalData>
```

## Ödeme İşlemi Mesaj Örneği

### Asp.Net

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<KuveytTurkVPosMessage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <APIVersion>1.0.0</APIVersion>
  <OkUrl>http://localhost:44785/Home/Success</OkUrl>
  <FailUrl>http://localhost:44785/Home/Fail</FailUrl>
  <HashData>yCb40Pd/fLlrX8GEhoamM9glKEc=</HashData>
  <MerchantId>80</MerchantId>
  <CustomerId>0</CustomerId>
  <UserName>apiuser</UserName>
  <CustomerIpAddress>1</CustomerIpAddress>
  <CardNumber>4025906425211421</CardNumber>
  <CardExpireDateYear>14</CardExpireDateYear>
  <CardExpireDateMonth>09</CardExpireDateMonth>
  <CardCVV2>154</CardCVV2>
  <CardHolderName>Koray Kirdinli</CardHolderName>
  <CardType>MasterCard</CardType>
  <TransactionType>Sale</TransactionType>
  <InstallmentCount>0</InstallmentCount>
  <Amount>72.56</Amount>
  <DisplayAmount>72,56</DisplayAmount>
  <CurrencyCode>0949</CurrencyCode>
  <MerchantOrderId>ORDER-123</MerchantOrderId>
  <TransactionSecurity>3</TransactionSecurity>
</KuveytTurkVPosMessage>
```

## Ödeme İşlemi Onayı Mesaj Örneği

### Asp.Net

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<KuveytTurkVPosMessage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <APIVersion>1.0.0</APIVersion>
  <HashData>yCb40Pd/fLlrX8GEhoamM9glKEc=</HashData>
  <MerchantId>80</MerchantId>
  <CustomerId>0</CustomerId>
```

```
<UserName>apiuser</UserName>
<CustomerIPAddress>1</CustomerIPAddress>
<AdditionalData>
  <KuveytTurkVPosAdditionalData>
    <Key>MD</Key>
    <Data>45646,45646,456464</Data>
    <Description></Description>
  </KuveytTurkVPosAdditionalData>
</AdditionalData>
<TransactionType>Sale</TransactionType>
<InstallmentCount>0</InstallmentCount>
<Amount>72.56</Amount>
<DisplayAmount>72,56</DisplayAmount>
<CurrencyCode>0949</CurrencyCode>
<MerchantOrderId>ORDER-123</MerchantOrderId>
<TransactionSecurity>3</TransactionSecurity>
</KuveytTurkVPosMessage>
```

## Ödeme İşlem Sonucu Mesaj Yapısı

### Asp.Net

```
<?xml version="1.0" encoding="utf-8"?>
<VPosTransactionResponseContract xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <VPosMessage>
    <OrderId>4480</OrderId>
    <OkUrl>http://localhost:10398/ThreeDModel/SuccessXml</OkUrl>
    <FailUrl>http://localhost:10398/ThreeDModel/FailXml</FailUrl>
    <MerchantId>1</MerchantId>
    <SubMerchantId>0</SubMerchantId>
    <CustomerId>873663357</CustomerId>
    <HashPassword>c77dFssAnYSy6O2MJ0+5tMYtGVc=</HashPassword>
    <CardNumber>4025893612740817</CardNumber>
    <BatchID>1906</BatchID>
    <InstallmentCount>0</InstallmentCount>
    <Amount>24</Amount>
    <MerchantOrderId>660723214</MerchantOrderId>
    <FECAmount>0</FECAmount>
    <CurrencyCode>949</CurrencyCode>
    <QeryId>0</QeryId>
    <DebtId>0</DebtId>
    <SurchargeAmount>0</SurchargeAmount>
    <SGKDebtAmount>0</SGKDebtAmount>
```



```
<TransactionSecurity>0</TransactionSecurity>
<PaymentId xsi:nil="true" />
<OrderPOSTransactionId xsi:nil="true" />
<TranDate xsi:nil="true"/>
</VPosMessage>
<IsEnrolled>true</IsEnrolled>
<ProvisionNumber>896626</ProvisionNumber>
<RRN>904115005554</RRN>
<Stan>005554</Stan>
<ResponseCode>00</ResponseCode>
<ResponseMessage>OTORİZASYON VERİLDİ</ResponseMessage>
<OrderId>4480</OrderId>
<TransactionTime>0001-01-01T00:00:00</TransactionTime>
<MerchantOrderId>660723214</MerchantOrderId>
<HashData>I7H/6nwfydM6VcwXsl82mqeC83o=</HashData>
</VPosTransactionResponseContract>
```

Not: Ödeme işlemine ait cevap mesajının içerisinde yer alan VPosMessage alanında, üye işyerlerinin ödeme işlemi için gönderdikleri mesaj yer almaktadır.(Bkz. "Ödeme İşlemi Mesaj yapısı")

Not: OkUrl veya FailUrl e iletilecek bu mesaj AuthenticationResponse altında UrlEncoded olacak şekilde gönderilmektedir. Mesajı kullanmak için gelen formun içinde AuthenticationResponse anahtarıyla gönderilen değeri UrlDecode ile çözünüz.( Örneğin .net için; System.Web.HttpUtility.UrlDecode(Request.Form["AuthenticationResponse"]) şeklinde kod parçası kullanınız.)

### HashData

Gelen cevabın Kuveyt Türk Sanal POS tarafından gönderildiğinin kontrol edilebilmesi için HashData değeri yer almaktadır. HashData alanının, üye işyerinin kendi oluşturduğu HashData ile aynı değere sahip olması bu cevap mesajının Kuveyt Türk Sanal POS tarafından gönderildiğini doğrulanmıştır.

Hash hesaplamasında kullanılan parametreler şöyledir: MerchantId, MerchantOrderId, Amount, OkUrl, FailUrl, UserName, hashPassword

## Örnek Kod Parçaları

### Asp.Net

```
public static String CreateHashString(KuveytTurkVPosMessage message, string hashPassword)
{
    string newHash = message.MerchantId + message.MerchantOrderId +
    message.Amount.ToAmountString() + message.OkUrl + message.FailUrl + message.UserName +
    hashPassword;
    return newHash;
}
```

```
public static String ComputeHash(String hashstr)
{
    System.Security.Cryptography.SHA1 sha = new
    System.Security.Cryptography.SHA1CryptoServiceProvider();
    byte[] hashbytes = System.Text.Encoding.GetEncoding("ISO-8859-9").GetBytes(hashstr);
    byte[] inputbytes = sha.ComputeHash(hashbytes);
    String hash = Convert.ToBase64String(inputbytes);
    return hash;
}

private static string DataPost(string uriToPost, string dataToPost)
{
    byte[] buffer = Encoding.UTF8.GetBytes(dataToPost);
    HttpWebRequest WebReq = (HttpWebRequest)WebRequest.Create(uriToPost);
    WebReq.Timeout = 5 * 60 * 1000;
    WebReq.Method = "POST";
    WebReq.ContentType = "application/xml";
    WebReq.ContentLength = buffer.Length;
    WebReq.CookieContainer = new CookieContainer();
    Stream ReqStream = WebReq.GetRequestStream();
    ReqStream.Write(buffer, 0, buffer.Length);
    ReqStream.Close();
    WebResponse WebRes = WebReq.GetResponse();
    Stream ResStream = WebRes.GetResponseStream();
    StreamReader ResReader = new StreamReader(ResStream);
    string responseString = ResReader.ReadToEnd();
    return responseString;
}
```

## Java

```
public static String post(String data, String url){

    try {
        URL u = new URL(url);
        URLConnection uc = u.openConnection();
        HttpURLConnection connection = (HttpURLConnection) uc;
        connection.setRequestProperty("Content-Type", "application/xml;
        charset=utf-8");
        connection.setDoOutput(true);
        connection.setDoInput(true);
        connection.setRequestMethod("POST");
        OutputStream out = connection.getOutputStream();
        OutputStreamWriter wout = new OutputStreamWriter(out, "UTF-8");
        wout.write(data);
        wout.flush();
        out.close();
        InputStream in = connection.getInputStream();
        String str = "";
        int c;
        while ((c = in.read()) != -1)
            str += (char)c;
        System.out.println("Res is: "+str);
        in.close();
        out.close();
        connection.disconnect();
    }
}
```

```
return str;
}
catch (IOException e) {
    System.err.println(e);
    e.printStackTrace();
    System.out.println();
    return null;
}

public static String calculateHash(String hashdata) {
    String returnText = "";
    MessageDigest sha1 = null;
    try {
        sha1 = MessageDigest.getInstance("SHA-1");
        sha1.update(hashdata.getBytes());
        returnText = B64ENC.encode(sha1.digest());
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    }
    return returnText;
}
```

### Php

```
try {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-type: application/xml', 'Content-length: '.
    strlen($xml)));
    curl_setopt($ch, CURLOPT_POST, true); //POST Metodu kullanarak verileri gönder
    curl_setopt($ch, CURLOPT_HEADER, false); //Serverdan gelen Header bilgilerini önemseme.
    curl_setopt($ch, CURLOPT_URL, 'https://boa.kuveytturk.com.tr/sanalposservice/Home/ThreeDModelPay
    Gate'); //Baglanacagi URL
    curl_setopt($ch, CURLOPT_POSTFIELDS, $xml);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true); //Transfer sonuçlarını al.
    $data = curl_exec($ch);
    curl_close($ch);
}
catch (Exception $e) {
    echo 'Caught exception: ', $e->getMessage(), "\n";
}
```

## Ödeme İşlemi Yapılması

### Asp.Net

```
[HttpPost]
public string Payment(VPosTransactionResponseContract model)
{
    ViewBag.Message = "";
    KuveytTurkVPosMessage request = new KuveytTurkVPosMessage();
    request.TransactionType = "Sale";
    request.CardNumber = model.VPosMessage.CardNumber;
```

```
request.CardExpireDateMonth = model.VPosMessage.CardExpireDateMonth;
request.CardExpireDateYear = model.VPosMessage.CardExpireDateYear;
request.CardType = model.VPosMessage.CardType;
request.CardCVV2 = model.VPosMessage.CardCVV2;
request.CardHolderName = model.VPosMessage.CardHolderName;
request.CurrencyCode = "0949";
request.APIVersion = "1.0.0";
request.MerchantId = merchantId;
request.CustomerId = customerId;
request.UserName = userName;
string password = hashedPassword;
request.MerchantOrderId = model.VPosMessage.MerchantOrderId;
request.Amount = model.VPosMessage.Amount;
request.DisplayAmount = request.Amount.ToString();
request.OkUrl = "http://localhost/MerchantTest/ThreeDModel/SuccessXml";
request.FailUrl = "http://localhost/MerchantTest/ThreeDModel/FailXml";
request.TransactionSecurity = model.VPosMessage.TransactionSecurity;
string newHash = Helper.CreateHashString(request, password);
request.HashData = Helper.ComputeHash(newHash);
XmlSerializer x = new XmlSerializer(request.GetType());
StringWriter sw = new StringWriter();
x.Serialize(sw, request);
XsdDataContractExporter exporter = new XsdDataContractExporter();
string str = sw.ToString();
str = sw.ToString();
str = str.Replace("encoding=\"utf-16\"", "encoding=\"ISO-8859-1\"");
string result = DataPost(PayGateUrl, str);
if (String.IsNullOrEmpty(result))
    result = "Boş Cevap";
return result;
}
```

## İşlem Sonucunun Alınması

### Asp.Net

```
private string GetPayloadResponse()
{
    string requestContent = string.Empty;
    requestContent = Request.Form["AuthenticationResponse"];
    requestContent = System.Web.HttpUtility.UrlDecode(requestContent);
    return requestContent;
}
```

## Ödeme İşleminin Onaylanması

### Asp.Net

```
[HttpPost]
public string Approve(VPosTransactionResponseContract model)
{
    KuveytTurkVPosMessage request = new KuveytTurkVPosMessage();
    request.MerchantId = merchantId;
    request.CustomerId = customerId;
    request.UserName = userName;
    string password = hashedPassword;

    request.MerchantOrderId = model.MerchantOrderId;
    request.Amount = 111;
```

```
string newHash = Helper.CreateHashString(request, password);
request.HashData = Helper.ComputeHash(newHash);
request.AdditionalData = new AdditionalData();
request.AdditionalData.AdditionalDataList = new
List<KuveytTurkVPosAdditionalData>();
KuveytTurkVPosAdditionalData md = new KuveytTurkVPosAdditionalData();
md.Key = "MD";
md.Data = model.MD;
request.AdditionalData.AdditionalDataList.Add(md);
XmlSerializer x = new XmlSerializer(request.GetType());
StringWriter sw = new StringWriter();
x.Serialize(sw, request);
XsdDataContractExporter exporter = new XsdDataContractExporter();
string str = sw.ToString();
str = sw.ToString();
str = str.Replace("encoding=\"utf-16\"", "encoding=\"ISO-8859-1\"");
string result = DataPost(ProvisionGate, str);
if (string.IsNullOrEmpty(result))
    result = "Boş Cevap";
return result;
}
```

### SanalPos 3D Model Ödeme Noktası Adresi (Production)

<https://boa.kuveytturk.com.tr/sanalposservice/Home/ThreeDModelPayGate>

### SanalPos 3D Model Ödeme Onaylama Adresi (Production)

<https://boa.kuveytturk.com.tr/sanalposservice/Home/ThreeDModelProvisionGate>

### SanalPos 3D Model Ödeme Noktası Adresi (Test)

<https://boatest.kuveytturk.com.tr/boa.virtualpos.services/Home/ThreeDModelPayGate>

### SanalPos 3D Model Ödeme Onaylama Adresi (Test)

<https://boatest.kuveytturk.com.tr/boa.virtualpos.services/Home/ThreeDModelProvisionGate>

### SanalPos 3D Model Ödeme Noktası Adresi (PreProduction)

<https://preprodsrv.kuveytturk.com.tr/sanalposservice/Home/ThreeDModelPayGate>

### SanalPos 3D Model Ödeme Onaylama Adresi (PreProduction)

<https://preprodsrv.kuveytturk.com.tr/sanalposservice/Home/ThreeDModelProvisionGate>