



# Sanal Pos

# API Entegrasyon

# Dökümanı

Vakıf Katılım Bankası

Versiyon 2.7

## İÇİNDEKİLER

<b>1. GİRİŞ.....</b>	<b>4</b>
<b>2. DESTEKLENEN İŞLEMLER .....</b>	<b>5</b>
<b>2.1. Kartlı Otorizasyon İşlemleri .....</b>	<b>5</b>
<b>2.1.1. Non3DSecure Ödeme İşlemi.....</b>	<b>5</b>
<b>2.1.2. 3DSecure Ödeme İşlemi .....</b>	<b>6</b>
<b>2.1.3. Mail Order İşlemi .....</b>	<b>8</b>
<b>2.1.4. Sigorta Şirketleri Mail Order İşlemi .....</b>	<b>8</b>
<b>2.1.5. İptal İşlemi .....</b>	<b>9</b>
<b>2.1.6. İade İşlemi.....</b>	<b>9</b>
<b>2.1.7. Kısmi İade İşlemi.....</b>	<b>9</b>
<b>2.1.8. Ön Provizyon İşlemi.....</b>	<b>10</b>
<b>2.1.9. Ön Provizyon Kapama İşlemi .....</b>	<b>10</b>
<b>2.1.10. Ön Provizyon Kısmi Kapama İşlemi .....</b>	<b>11</b>
<b>2.1.11. Ön Provizyon İptal İşlemi .....</b>	<b>11</b>
<b>2.1.12. Ön Provizyon İade İşlemi .....</b>	<b>11</b>
<b>2.2. Saklanan Kart Verisi İşlemleri .....</b>	<b>12</b>
<b>2.2.1. CardSave İşlemi .....</b>	<b>12</b>
<b>2.2.2. CardSelect İşlemi .....</b>	<b>12</b>
<b>2.2.3. CardSelectAll İşlemi.....</b>	<b>12</b>
<b>2.2.4. CardDelete İşlemi .....</b>	<b>13</b>
<b>2.3. Sipariş İşlemleri.....</b>	<b>13</b>
<b>2.3.1. SelectOrder İşlemi.....</b>	<b>13</b>
<b>2.3.2. SelectOrderByMerchantOrderId İşlemi.....</b>	<b>14</b>
<b>3. MESAJ YAPILARI VE ÖRNEK MESAJLAR / KODLAR.....</b>	<b>14</b>
<b>3.1. Kartlı İşlemler Mesaj Yapıları.....</b>	<b>14</b>
<b>3.1.1. Non3DSecure Ödeme İşlemi.....</b>	<b>14</b>
<b>3.1.2. 3DSecure Ödeme İşlemi .....</b>	<b>18</b>
<b>3.1.3. Mail Order İşlemi .....</b>	<b>29</b>
<b>3.1.4. Sigorta Şirketleri Mail Order İşlemi .....</b>	<b>35</b>
<b>3.1.5. İptal İşlemi .....</b>	<b>40</b>

3.1.6. İade İşlemi.....	43
3.1.7. Kısmi İade İşlemi .....	47
3.1.8. Ön Provizyon İşlemi.....	50
3.1.9. Ön Provizyon Kapama İşlemi .....	56
3.1.10. Ön Provizyon Kısmi Kapama İşlemi .....	59
3.1.11. Ön Provizyon İptal İşlemi .....	63
3.1.12. Ön Provizyon İade İşlemi .....	67
3.2. Saklanan Kart İşlemleri Mesaj Yapıları .....	70
3.2.1. CardSave İşlemi .....	70
3.2.2. CardSelect İşlemi .....	74
3.2.3. CardSelectAll İşlemi.....	77
3.2.4. CardDelete İşlemi .....	80
3.3. Sipariş İşlemleri Mesaj Yapıları .....	82
3.3.1. SelectOrder İşlemi.....	82
3.3.2. SelectOrderByMerchantOrderId İşlemi.....	87
4. GEREKLİ BİLGİLER .....	90
4.1. Değişken Tipleri .....	90
4.2. Model Sınıfları.....	94
4.3. Örnek Kodlarda ki Yardımcı Methodlar.....	97
4.4. Hash Data Hesaplaması.....	98
4.4.1. Hash data hesaplamalarında kullanılacak methodlar:.....	98
4.4.2. Hash Data Hesaplamasında Hata Alınması Halinde.....	99
4.5. Api Servisleri İstek Gönderilmesi ve Geri Dönен Cevabın Alınması .....	99
4.5.1. İstek Post Edilmesi için Method .....	99
4.5.2. Cevabın Alınması .....	100
4.6. Cevap Kodlarının Okutulduğu Alanlar .....	101
4.6.1. Cevap Kodlarının Açıklamaları.....	101

## 1. GİRİŞ

Bu doküman, Vakıf Katılım Sanal POS sistemine entegre olacak kullanıcılar için gerekli olan teknik detayları, mesaj yapılarını ve örneklerini ve ihtiyaç duyulabilecek diğer bilgileri içermektedir.

**Önemli Bilgi:** **Api entegrasyon adımlarına geçmeden önce, api servislerini kullanmak için sanal pos kurumsal sitesi üzerinden api kullanıcısı tanımı yapmanız gerekmektedir.** Bu adım, “Sanal POS Kullanım Kılavuzu” dökümanında detaylı şekilde anlatılmaktadır. Oluşturduğunuz şifreyi bu dökümda yer alan hash standartlarına uygun şekilde hashleyip, api kullanıcı adınızı “Username” ve hashli şifreyi “HashPassword” alanlarına set edip göndermelisiniz.

Dikkat etmeniz gereken diğer bir husus ise, api servislerini kullanabilmek için erişim gerçekleştireceğiniz makine IP adresinin tanımlı olması gerekmektedir. Bu adımlarla ilgili destek almak için [sanalposdestek@vakifkatilim.com.tr](mailto:sanalposdestek@vakifkatilim.com.tr) adresi vasıtıyla iletişim kurabilirsiniz.

## 2. DESTEKLENEN İŞLEMLER

### 2.1. Kartlı Otorizasyon İşlemleri

#### 2.1.1. Non3DSecure Ödeme İşlemi

Ödeme işlemi, kullanıcıların 3D Secure güvenlik sistemi olmadan daha hızlı ödeme yapmasına olanak sağlayan işlem tipidir. Ödeme işlemiyle ilgili bazı önemli bilgiler aşağıdaki gibidir:

- 3D Secure ödeme sistemlerinin sunduğu avantajları sağlamamaktadır.
- Non3D Secure sistemi uygulanan ödeme işlemlerinde, kullanıcıdan kart bilgileri alındıktan sonra ödeme işlemi yapılır.
- Ödeme işleminin gerçekleşebilmesi için kartın E-commerce işlemlere açık olması yeterlidir.

#### Non3DSecure İşlemde Kullanılacak Bilgiler

Üye işyerinin Non3D Secure ödeme uygulayabilmesi için Madde 4.1 de yer alan Değişken Tiplerinde belirtilen alanlara ve XML desenine uygun şekilde istek göndermeleri gerekmektedir.(Bkz. [Non3DSecure Ödeme İşlemi](#) ve [Değişken Bilgileri](#))

#### Non3DSecure İşlem Akışı

SanalPos tarafından sunulan Non3D Secure ödeme yöntemi için kullanılacak örnek bir entegrasyon işlemi bu kısımda anlatılmaktadır:

1. Üye işyeri sipariş detayını görüntüler.
2. Kullanıcı,
  - a. Kart Numarası,
  - b. Son Kullanım Tarihi
  - c. CVV bilgilerini girer.
3. 3D Secure ödeme işlem seçenekini seçmeden işleme devam eder ya da sanal işyeri işlemi 3D secure akışına yönlendirmeden işleme devam eder.

4. Üye işyeri müşteriden almış olduğu bilgilerle Kart, sipariş ve üye işyerine ait bilgileri içeren mesaj oluşturulur. Mesajda yer alacak HashData alanına dikkat edilmelidir. Bu mesaj yapısı XML formatındadır.(Bkz. "[Non3DSecure Ödeme İşlemi](#)")
5. Üye işyeri oluşturduğu XML mesajı SanalPos tarafından sunulan Non3D Secure ödeme noktasına HTTP POST yöntemi ile gönderilir. Bu işlem sonucunda XML yapısında bir cevap yer almaktadır. Üye işyeri aldığı bu cevabı işleyip işlemlerine devam etmelidir.(Bkz. "[Non3DSecure Ödeme İşlemi](#)", "[Non3DSecure Ödeme İşlemi Adresi](#)")
6. Gelen XML cevapta ResponseCode değeri "00" olduğu durumda ödeme işlemi gerçekleşmiş olmaktadır.

### Non3DSecure Ödeme İşlemi Adresi

<https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/Non3DPayGate>

#### 2.1.2. 3DSecure Ödeme İşlemi

3D Secure, İnternet üzerinden kredi kartı ve banka kartıyla yapılan alışveriş işlem güvenliğinin artırılması için Mastercard ve Visa tarafından geliştirilen bir sistemdir.

- 3D Secure sistemi kullanıcı, üye işyeri ve banka arasındaki sorumlulukları düzenlemektedir.
- 3D Secure olarak yapılan ödeme işlemlerinde kullanıcı;
  - ekrandan kart bilgilerini girer
  - bankanın 3D Secure ödeme sayfasına yönlendirilir.
  - banka tarafından kendisine gönderilen kısa mesajda yer alan şifreyi girer,
- Girilen şifre doğru ise, kart doğrulaması yapılarak işlem kullanıcı tarafından imzalanmış demektir.
- 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 Secure uygulanan standart ödeme işleminde aşağıdaki temel adımlar gerçekleştirilir:
  - Kullanıcının karta ait bilgileri girmesi

- Kart sahibi banka tarafından kartın doğrulanması
- Kullanıcının işlemi imzalaması
- Doğrulama sonucunun üye işyeri tarafından kontrol edilmesi
- Ödemenin gerçekleştirilmesi

### **3DSecure İşlemde Kullanılacak Bilgiler**

Üye işyerinin 3D Secure ödeme uygulayabilmesi için Madde 4.1 de yer alan Değişken Tiplerinde belirtilen alanlara ve XML desenine uygun şekilde istek göndermeleri gerekmektedir.(Bkz. [3DSecure Mesaj Yapısı](#) , [Ödeme Onayı Mesaj yapısı](#) ve [Değişken Bilgileri](#))

### **3DSecure İşlem Akışı**

Sanal Pos tarafından sunulan 3D Secure ödeme yöntemi için kullanılacak örnek bir entegrasyon işlemi bu kısımda anlatılmaktadır:

1. Üye işyeri sipariş detayını görüntüler.
2. Kullanıcı,
  - Kart Numarası
  - Son Kullanım Tarihi
  - CVV bilgilerini girer.
3. 3D Secure ödeme işlem seçeneğini seçer ya da sanal işyeri işlemi otomatik olarak 3D secure akışına yönlendirir.
4. Üye işyeri müşteriden almış olduğu bilgilerle Kart, sipariş ve üye işyerine ait bilgileri içeren mesaj oluşturulur. Mesajda yer alacak HashData alanına dikkat edilmelidir. Bu mesaj yapısı XML formatındadır.(Bkz. “[3DSecure Ödeme İşlemi](#)”)
5. Üye işyeri oluşturduğu XML mesajı SanalPos tarafından sunulan 3D Secure ödeme noktasına HTTP POST yöntemi ile gönderir. Bu işlem sonucunda HTML yapısında bir cevap yer almaktadır. Üye işyeri bu HTML cevabı tarayıcıya iletmelidir.(Bkz. “[3DSecure Ödeme İşlemi](#)”, “[3DSecure Ödeme İşlemi Adresi](#)”)
6. Sanal Postan gerçekleştirilen işlemlere ait sonuç bilgileri XML mesaj şeklinde FailUrl ya da OkUrl e HTTP Post ile iletilecektir. (Bkz. “[3DSecure Ödeme İşlemi](#)”)

7. 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 Sanal Pos 3D Secure Provizyon adresine ödeme mesajı göndermelidir. Gönderilecek mesajda, bu noktada gelen cevapta yer alan "MD" değeri ile birlikte ilk aşamada gönderilen üye işyeri ve sipariş bilgileri gönderilmelidir. Sanal Pos 3D Secure Provizyon adresine gönderilen mesaj Sanal Pos servisi tarafından işlenir ve XML cevap verilir. XML cevapta yer alan ResponseCode değeri "00" olduğu durumda ödeme işlemi gerçekleşmiş olmaktadır.(Bkz. [3DSecure Ödeme İşlemi](#), [3DSecure Ödeme İşlemi Adresi](#))

### **3DSecure Ödeme İşlemi Adresi**

<https://boa.vakifkatlim.com.tr/VirtualPOS.Gateway/Home/ThreeDModelPayGate>

### **3DSecure Ödeme Onaylama Adresi**

<https://boa.vakifkatlim.com.tr/VirtualPOS.Gateway/Home/ThreeDModelProvisionGate>

#### **2.1.3. Mail Order İşlemi**

Mail order işlemi, ödeme yapacak kartın fiziken pos cihazının yanında olmadan, kart bilgilerinin (kart no, son kullanım tarihi ve CVV ) pos cihazı üzerine tuşlanması yoluyla yapılan işlemidir.

Madde 4.1 de yer alan Değişken Tiplerinde belirtilen alanlara ve XML desenine uygun şekilde istek gönderilerek işlem gerçekleştirilebilir.(Bkz. [Mail Order Mesaj Yapısı](#) ve [Değişken Bilgileri](#))

### **Mail Order İşlemi Adresi**

<https://boa.vakifkatlim.com.tr/VirtualPOS.Gateway/Home/MailOrderSale>

#### **2.1.4. Sigorta Şirketleri Mail Order İşlemi**

Sigorta şirketlerinin sanal pos entegrasyonunun sağlanması ile birlikte bu sektörde ait firmaların sanal pos işlemleri de mail order işlemi olarak gerçekleşecektir. Normal mail order işleminden

farklı olarak burada kart numarasının ilk 6 ve son 4 haneleri ile (Örnek: 123456\*7890) TCKN/VKN bilgileri ile satış işlemleri gerçekleştirilecektir.

Değişken tiplerinde belirtilen alanlara ve XML desenine uygun şekilde istek gönderilerek işlem gerçekleştirilebilir.(Bkz. [Mail Order Mesaj Yapisı](#) ve [Değişken Bilgileri](#))

### Sigorta Şirketleri Mail Order İşlemi Adresi

<https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/MailOrderSale>

#### 2.1.5. İptal İşlemi

Gün sonu alınmamış satış işlemlerinin iptalinin yapılmasına olanak sağlayan işlemidir.

#### İptal İşleminde Kullanılacak Bilgiler

Madde 4.1 de yer alan Değişken Tiplerinde belirtilen alanlara ve XML desenine uygun şekilde istek gönderilerek işlem gerçekleştirilebilir.(Bkz. [İptal Mesaj Yapisı](#) ve [Değişken Bilgileri](#))

#### İptal İşlemi Adresi

<https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/SaleReversal>

#### 2.1.6. İade İşlemi

Gün sonu alınmış satış işlemlerinin iade yapılmasına olanak sağlayan işlemidir.

Madde 4.1 de yer alan Değişken Tiplerinde belirtilen alanlara ve XML desenine uygun şekilde istek gönderilerek işlem gerçekleştirilebilir.(Bkz. [İade Mesaj Yapisı](#) ve [Değişken Bilgileri](#))

#### İade İşlemi Adresi

<https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/DrawBack>

#### 2.1.7. Kısmi İade İşlemi

Gün sonu alınmış satış işlemlerinin kısmi iadesinin yapılmasına olanak sağlayan işlemidir.

Madde 4.1 de yer alan Değişken Tiplerinde belirtilen alanlara ve XML desenine uygun şekilde istek gönderilerek işlem gerçekleştirilebilir.(Bkz. [Kısmi İade Mesaj Yapısı](#) ve [Değişken Bilgileri](#))

### Kısmi İade İşlemi Adresi

<https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/PartialDrawBack>

#### 2.1.8. Ön Provizyon İşlemi

İleri tarihli yapılacak olan ürün ya da hizmet satışında müşterinin kredi kartının geçerliliğini ve harcama için yeterli limiti olup olmadığını kontrol etmek için kullanılır. Genellikle rezervasyon işlemleri için kullanılan bir yöntemdir.

Madde 4.1 de yer alan Değişken Tiplerinde belirtilen alanlara ve XML desenine uygun şekilde istek gönderilerek işlem gerçekleştirilebilir.(Bkz. [Ön Provizyon İşlemi](#) ve [Değişken Bilgileri](#))

### Ön Provizyon İşlemi Adresi

<https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/PreAuthorizaten>

#### 2.1.9. Ön Provizyon Kapama İşlemi

Ön Provizyon ile belirlenen limitin karta hareket olarak yansımmasını sağlar. Ön Provizyon kapama işlemi, standart Satış işlemi gibi değerlendirilebilir.

Madde 4.1 de yer alan Değişken Tiplerinde belirtilen alanlara ve XML desenine uygun şekilde istek gönderilerek işlem gerçekleştirilebilir.(Bkz. [Ön Prov Kapama Mesaj Yapısı](#) ve [Değişken Bilgileri](#))

### Ön Provizyon Kapama İşlemi Adresi

<https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/PreAuthorizationClose>

### 2.1.10. Ön Provizyon Kısmi Kapama İşlemi

Ön Provizyon ile belirlenen limitin kısmen karta hareket olarak yansımmasını sağlar. Ön Provizyon kısmı kapama işlemi, standart Satış işlemi gibi değerlendirilebilir.

Madde 4.1 de yer alan Değişken Tiplerinde belirtilen alanlara ve XML desenine uygun şekilde istek gönderilerek işlem gerçekleştirilebilir.(Bkz. Ön Prov Kısmi Kapama Mesaj Yapısı ve Değişken Bilgileri)

#### Ön Provizyon Kısmi Kapama İşlemi Adresi

<https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/PreAuthorizationPartialClose>

### 2.1.11. Ön Provizyon İptal İşlemi

Gün sonu alınmamış ön provizyon işlemlerinin iptalinin yapılabilmesi sağlanmaktadır.

Madde 4.1 de yer alan Değişken Tiplerinde belirtilen alanlara ve XML desenine uygun şekilde istek gönderilerek işlem gerçekleştirilebilir.(Bkz. Ön Prov İptal Mesaj Yapısı ve Değişken Bilgileri)

#### Ön Provizyon İptal İşlemi Adresi

<https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/PreAuthorizationReversal>

### 2.1.12. Ön Provizyon İade İşlemi

Gün sonu alınmış ön provizyon işlem iptallerinin yapılması sağlanmaktadır.

Madde 4.1 de yer alan Değişken Tiplerinde belirtilen alanlara ve XML desenine uygun şekilde istek gönderilerek işlem gerçekleştirilebilir.(Bkz. Ön Prov İade Mesaj Yapısı ve Değişken Bilgileri)

#### Ön Provizyon İade İşlemi Adresi

<https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/PreAuthorizationDrawBack>

## 2.2. Saklanan Kart Verisi İşlemleri

### 2.2.1. CardSave İşlemi

Kart saklama işlemlerinde kart bilgilerinin şifreli şekilde güvenli olarak saklanması sağlanmaktadır.

Servis kart bilgileri ile çağrııldığında, o bilgileri temsil eden bir cardguid döner ve üye işyeri satış işlemini sadece o CardGuid değeri ile gerçekleştirebilir.

Sigorta firmaları ise istek içerisinde yer alan cvv2 ve expire date alanları boş geçip IdentityTaxNumber ve CardNumber alanında 6-4 maskeli kart verisi göndererek işlemlerini gerçekleştireceklerdir. Sigorta firması dışında yer alan üye işyerleri IdentityTaxNumber boş göndermeleri gerekmektedir.

Madde 4.1 de yer alan Değişken Tiplerinde belirtilen alanlara ve XML desenine uygun şekilde istek gönderilerek işlem gerçekleştirilebilir.(Bkz. [CardSave İşlemi](#) ve [Değişken Bilgileri](#))

#### CardSave İşlemi Adresi

<https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/CardSave>

### 2.2.2. CardSelect İşlemi

Şifreli bir şekilde saklanan kart verisinin şifresinin çözülmerek gösterilmesi sağlanmaktadır.

Servis ilgili saklı kart verisinin CardGuide değeri ile çağrııldığında ilgili servis kart sahibi adı soyadı, kart numarası alanını 4-4 maskeli kart numarsı olarak ve sistem tarihi ile döner.

Madde 4.1 de yer alan Değişken Tiplerinde belirtilen alanlara ve XML desenine uygun şekilde istek gönderilerek işlem gerçekleştirilebilir.(Bkz. [CardSelect İşlemi Mesaj Yapısı ve Örnek Mesaj](#) ve [Değişken Bilgileri](#))

#### CardSelect İşlemi Adresi

<https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/CardSelect>

### 2.2.3. CardSelectAll İşlemi

Şifreli bir şekilde saklanan kart verilerinin belirli tarih aralığında şifresinin çözülmerek gösterilmesi sağlanmaktadır.

Servis ilgili saklı kart verilerinin kayıt tarihine göre başlangıç ve bitiş tarihleri ile çağrııldığında ilgili servis kart sahibi adı soyadı, kart numarası alanını 4-4 maskeli kart numarsı olarak ve sistem tarihi ile belirtilen tarih aralığında kaydedilmiş kart bilgilerini listeler.

Madde 4.1 de yer alan Değişken Tiplerinde belirtilen alanlara ve XML desenine uygun şekilde istek gönderilerek işlem gerçekleştirilebilir.(Bkz. [CardSelectAll İşlemi Mesaj Yapısı ve Örnek Mesaj](#) ve [Değişken Bilgileri](#))

### **CardSelectAll İşlemi Adresi**

<https://boa.vakifkatlim.com.tr/VirtualPOS.Gateway/Home/CardSelectAll>

#### **2.2.4. CardDelete İşlemi**

Şifreli bir şekilde saklanan kart verilerinin pasif hale getirilmesini sağlamaktadır.

Servis ilgili saklı kart verisinin CardGuide değeri ile çağrııldığında ilgili servis gönderilen kart verisini tablodan IsActive değerini 0 değerine getirerek pasif hale getirir.

Madde 4.1 de yer alan Değişken Tiplerinde belirtilen alanlara ve XML desenine uygun şekilde istek gönderilerek işlem gerçekleştirilebilir.(Bkz. [CardDelete İşlemi Mesaj Yapısı ve Örnek Mesaj](#) ve [Değişken Bilgileri](#))

### **CardDelete İşlemi Adresi**

<https://boa.vakifkatlim.com.tr/VirtualPOS.Gateway/Home/CardDelete>

## **2.3. Sipariş İşlemleri**

### **2.3.1. SelectOrder İşlemi**

Üye işyerinin tarih aralığı ve buna ek olarak tutar aralığı vererek ya da direkt onaylı işlemin provizyon numarası ile sanal pos üzerinden geçen işlemlerini çekebildiği api servisidir. Bunlara ek olarak sipariş durumunu ifade eden sipariş satatüsü, işlemin sonucunu ifade eden işlem sonucu ve sipariş numarası alanları ile de işlem çekilebilecektir. Başlangıç ve bitiş tarihi alanları zorunlu olup, limit alanları, provizyon numarası, sipariş statüsü, işlem sonucu, sipariş numarası alanları opsioneldir. Tarih aralığı maksimum 90 gün olabilir ve ilgili kritelere uygun herhangi bir veri yoksa veri yok şeklinde cevap döner.

Madde 4.1 de yer alan Değişken Tiplerinde belirtilen alanlara ve XML desenine uygun şekilde istek gönderilerek işlem gerçekleştirilebilir.(Bkz. [SelectOrder İşlemi](#) ve [Değişken Bilgileri](#))

## SelectOrder İşlemi Adresi

[### 2.3.2. SelectOrderByMerchantOrderId İşlemi](https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home>SelectOrder</a></p></div><div data-bbox=)

Üye işyerinin, daha önce günderiği üye işyeri sipariş numarası ile ilgili sipariş verilerinin dönüldüğü api servisidir. Bu servisi kullanırken Üye işyeri sipariş numarasının her ödeme işlemi için tekil olarak gönderilmesi önerilmektedir. Ayrıca sanal pos destek ekibinden üye işyeri sipariş numarasının tekil olup olmadığı kontrolü tanımlanabilir. Bu sayede eğer daha önce gönderilen ve başarılı sonuçlanan bir üye işyeri sipariş numarası ile tekrar ödeme isteği gönderilse bu işlem pos host tarafında reddedilir.

Madde 4.1 de yer alan Değişken Tiplerinde belirtilen alanlara ve XML desenine uygun şekilde istek gönderilerek işlem gerçekleştirilebilir. (Bkz. [SelectOrderByMerchantOrderId İşlemi](#) ve [Değişken Bilgileri](#))

## SelectOrderByMerchantOrderId İşlemi Adresi

[## 3. MESAJ YAPILARI VE ÖRNEK MESAJLAR / KODLAR](https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home>SelectOrderByMerchantOrderId</a></p></div><div data-bbox=)

### 3.1. Kartlı İşlemler Mesaj Yapıları

#### 3.1.1. Non3DSecure Ödeme İşlemi

##### Ödeme İşlemi Mesaj Yapısı

```
<VPosMessageContract>
  <HashData> </HashData>
  <MerchantId></MerchantId>
  <CustomerId></CustomerId>
  <UserName> </UserName>
  <CustomerIPAddress></CustomerIPAddress>
  <MerchantOrderId></MerchantOrderId>
  <InstallmentCount></InstallmentCount>
  <Amount></Amount>
  <DisplayAmount></DisplayAmount>
```

```
<FECCurrencyCode></FECCurrencyCode>
<CurrencyCode></CurrencyCode>
<Addresses>
    <VPosAddressContract>
        <Type></Type>
        <Name> </Name>
        <PhoneNumber></PhoneNumber>
        <OrderId></OrderId>
        <AddressId></AddressId>
        <Email></Email>
    </VPosAddressContract>
</Addresses>
<CardNumber></CardNumber>
<CardExpireDateYear></CardExpireDateYear>
<CardExpireDateMonth></CardExpireDateMonth>
<CardCVV2></CardCVV2>
<CardHolderName> </CardHolderName>
<PaymentType> </PaymentType>
<TransactionSecurity></TransactionSecurity>
</VPosMessageContract>
```

### Ödeme İşlemi Örnek Mesaj

```
<?xml version="1.0" encoding="UTF-16"?>
<VPosMessageContract
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <HashData>kAzwrExT6hFy7TyHMZFCp4a9ow0=</HashData>
    <MerchantId>1</MerchantId>
    <CustomerId>11111</CustomerId>
    <UserName>APIUSER</UserName>
    <HashPassword>kfkdsnskslkclswr9430ır</ HashPassword >
```

```
<MerchantOrderId>1517747916</MerchantOrderId>
<InstallmentCount>0</InstallmentCount>
<Amount>360</Amount>
<DisplayAmount>360</DisplayAmount>
<FECCurrencyCode>0949</FECCurrencyCode>
<CurrencyCode>0949</CurrencyCode>
<Addresses>
    <VPosAddressContract>
        <Type>1</Type>
        <Name>Mahmut Sami YAZAR</Name>
        <PhoneNumber>324234234234</PhoneNumber>
        <OrderId>0</OrderId>
        <AddressId>12</AddressId>
        <Email>mahmutsamiyazar@hotmail.com</Email>
    </VPosAddressContract>
</Addresses>
<CardNumber>5353550000958906</CardNumber>
<CardExpireDateYear>23</CardExpireDateYear>
<CardExpireDateMonth>01</CardExpireDateMonth>
<CardCVV2>741</CardCVV2>
<CardHolderName>Hasan Karacan</CardHolderName>
<PaymentType>1</PaymentType>
<TransactionSecurity>1</TransactionSecurity>
</VPosMessageContract>
```

### Ödeme İşlemi Sonucu Mesaj Yapısı

```
< VPosMessageContract >
    < VPOSMessage></VPOSMessage>
    <IsEnrolled></IsEnrolled>
    <IsVirtual> </IsVirtual>
```

```
<ProvisionNumber></ProvisionNumber>
<RRN></RRN>
<Stan></Stan>
<ResponseCode></ResponseCode>
<ResponseMessage> </ResponseMessage>
<OrderId></OrderId>
<TransactionTime></TransactionTime>
<MerchantOrderId></MerchantOrderId>
<HashData></HashData>
<BusinessKey></BusinessKey>
</ VPosMessageContract >
```

## Non3DSecure Ödeme İşlemi Örnek Kodlar

### Ödeme İşlemi Yapılması

#### Asp.Net

```
[HttpPost]
public VPosTransactionResponseContract Non3dPayment()
{
    VPosMessageContract request = new VPosMessageContract();
    request.PaymentType = 1;
    request.CardNumber = "";
    request.CardExpireDateMonth = "";
    request.CardExpireDateYear = "";
    request.CardCVV2 = "";
    request.CardHolderName = "";
    request.FECCurrencyCode = "0949";
    request.CurrencyCode = "0949";
    request.APIVersion = "";
    request.MerchantId = MerchantId;
    request.CustomerId = CustomerId;
    request.UserName = "";
    request.MerchantOrderId = "";
    request.InstallmentCount = 0;
    request.Amount = 1;
    request.DisplayAmount = "100";
    request.Addresses = new List<VPosAddressContract>();
    VPosAddressContract adres = new VPosAddressContract();
    adres.Name = "isminiz";
    adres.PhoneNumber = "telefon numaranız";
    adres.Email = "mail adresiniz";
    adres.AddressId = 0;
```

```

    adres.Type = "";
    request.Addresses.Add(adres);
    request.TransactionSecurity = 1;

    //HashData Hesaplanması
    string hashPassword = "";
    string hashString = CreateHashString(merchantId: request.MerchantId, merchantOrderId:
    "", amount: default(int), okUrl: "", failUrl: "", username: request.UserName,
    hashPassword: hashPassword);
    request.HashData = ComputeHash(hashString);

    VPosTransactionResponseContract response = new VPosTransactionResponseContract();

    XmlSerializer x = new XmlSerializer(request.GetType());
    StringWriter sw = new StringWriter();
    x.Serialize(sw, request);
    XsdDataContractExporter exporter = new XsdDataContractExporter();
    string requestMessage = sw.ToString();

    string result =
    DataPost("https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/Non3DPayGate",
    requestMessage);
    try
    {
        response =
        DeserializeObject<VPosTransactionResponseContract>(result.Replace("\r\n", "").Replace(
        encoding="utf-16\"", ""));
    }
    catch (Exception ex)
    {
        string errorMessage = "Mesaj1: " + ex.Message + ": Mesaj2: " + ex.StackTrace + ":" +
        Mesaj3: " + ex.InnerException;
    }

    return response;
}

```

### 3.1.2. 3DSecure Ödeme İşlemi

#### Ödeme İşlemi

##### Ödeme İşlemi Mesaj Yapısı

###### Asp.Net

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```



```
<VPosMessageContract
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <OkUrl> </OkUrl>
  <FailUrl> </FailUrl>
  <HashData> </HashData>
  <MerchantId></MerchantId>
  <SubMerchantId></SubMerchantId>
  <CustomerId></CustomerId>
  <UserName></UserName>
  <MerchantOrderId></MerchantOrderId>
  <InstallmentCount></InstallmentCount>
  <Amount></Amount>
  <DisplayAmount></DisplayAmount>
  <FECAMount></FECAMount>
  <FECCurrencyCode></FECCurrencyCode> <AdditionalData>
    <AdditionalDataList>
      <VPosAdditionalData>
        <Key> </Key>
        <Data> </Data>
        <Description> Ek Bilgi </Description>
      </VPosAdditionalData>
    </AdditionalDataList>
  </AdditionalData>
  <Addresses>
    <VPosAddressContract>
      <Type></Type>
      <Name> </Name>
      <PhoneNumber></PhoneNumber>
      <OrderId></OrderId>
      <AddressId></AddressId>
      <Email> Email</Email>
    </VPosAddressContract>
  
```

```
</Addresses>
<APIVersion></APIVersion>
<CardNumber></CardNumber>
<CardExpireDateYear></CardExpireDateYear>
<CardExpireDateMonth></CardExpireDateMonth>
<CardCVV2></CardCVV2>
<CardHolderName></CardHolderName>
<PaymentType></PaymentType>
<SurchargeAmount></SurchargeAmount>
<SGKDebtAmount></SGKDebtAmount>
<InstallmentMaturityCommisionFlag></InstallmentMaturityCommisionFlag>
<TransactionSecurity></TransactionSecurity>
</VPosMessageContract>
```

## Ödeme İşlemi Mesaj Örneği

### Asp.Net

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<VPosMessageContract
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <OkUrl>http://localhost:10398/ThreeDModel/Success</OkUrl>
  <FailUrl>http://localhost:10398/ThreeDModel/Fail</FailUrl>
  <HashData>yif9oJWlu6dWdwPhB6enwKbL9Sc=</HashData>
  <MerchantId>1</MerchantId>
  <SubMerchantId>0</SubMerchantId>
  <CustomerId>11111</CustomerId>
  <UserName>APIUSER</UserName>
  <HashPassword>kfkdsnskslkclswr9430ır</HashPassword>
  <MerchantOrderId>493702523</MerchantOrderId>
  <InstallmentCount>0</InstallmentCount>
  <Amount>30</Amount>
  <DisplayAmount>30</DisplayAmount>
```

```
<FECAmount>0</FECAmount>
<FECCurrencyCode>0949</FECCurrencyCode> <AdditionalData>
    <AdditionalDataList>
        <VPosAdditionalData>
            <Key>keyasdadsads</Key>
            <Data>asadasdsad</Data>
            <Description>descdsadasd</Description>
        </VPosAdditionalData>
    </AdditionalDataList>
</AdditionalData>
<Addresses>
    <VPosAddressContract>
        <Type>1</Type>
        <Name>Mahmut YAZAR</Name>
        <PhoneNumber>324234234234</PhoneNumber>
        <OrderId>0</OrderId>
        <AddressId>12</AddressId>
        <Email>mahmut@hotmail.com</Email>
    </VPosAddressContract>
</Addresses>
<APIVersion>1.0.0</APIVersion>
<CardNumber>5353550000958906</CardNumber>
<CardExpireDateYear>23</CardExpireDateYear>
<CardExpireDateMonth>01</CardExpireDateMonth>
<CardCVV2>741</CardCVV2>
<CardHolderName>Hasan Karacan</CardHolderName>
<PaymentType>1</PaymentType>
<DebtId>0</DebtId>
<SurchargeAmount>0</SurchargeAmount>
<SGKDebtAmount>0</SGKDebtAmount>
<InstallmentMaturityCommisionFlag>0</InstallmentMaturityCo mmisionFlag>
<TransactionSecurity>3</TransactionSecurity>
</VPosMessageContract>
```

## Ödeme İşlemi Sonucu Mesaj Yapısı

3d secure işleminde ilk adım da kart ve alıcı doğrulaması yapılır. Bu isteğin sonucu olarak bir html sayfa döner bunu ortamınızda bir web sayfası gibi çalıştırmanız. Bu şekilde ilgili banka şifre giriş ekranına yönlenmiş olur.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<title></title>
</head>

<body onload="OnLoadEvent();">
<form name="downloadForm" action="https://localhost/VirtualPos/ThreeDModel/Fail"
method="POST">
<input type="hidden" name="ResponseCode" value="CardNotEnrolled">
<input type="hidden" name="ResponseMessage" value="Card 3D Secure kayitli degil.">
<input type="hidden" name="ProvisionNumber">
<input type="hidden" name="MerchantOrderId">
<input type="hidden" name="OrderId" value="0">
<input type="hidden" name="RRN">
<input type="hidden" name="Stan">
<input type="hidden" name="HashData">
<input type="hidden" name="MD">
<!-- To support javascript unaware/disabled browsers -->
<noscript>
<center>
Please click the submit button below.<br>
<input type="submit" name="submit" value="Submit">
</center>
</noscript>
</form>
```

```
<script language="Javascript">
  <!--
  function OnLoadEvent() {
    document.downloadForm.submit();
  }
  //
  -->
</script>
</body>

</html>
```

## Ödeme Onayı

### Ödeme Onayı Mesaj yapısı

#### Asp.Net

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<VPosMessageContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <OkUrl> </OkUrl>
  <FailUrl> </FailUrl>
  <HashData> </HashData>
  <MerchantId></MerchantId>
  <SubMerchantId></SubMerchantId>
  <BatchId></BatchId>
  <MerchantOrderId></MerchantOrderId>
  <InstallmentCount></InstallmentCount>
  <Amount></Amount>
  <FECAmount></FECAmount>
  <AdditionalData>
    <AdditionalDataList>
      <VPosAdditionalData>
```

```
<Key></Key>
<Data> </Data>
</VPosAdditionalData>
</AdditionalDataList>
</AdditionalData>
<PaymentType> </PaymentType>
<SurchargeAmount></SurchargeAmount>
<SGKDebtAmount></SGKDebtAmount>
<InstallmentMaturityCommisionFlag></InstallmentMaturityCommisionFlag>
<TransactionSecurity></TransactionSecurity>
<\VPosMessageContract>
```

## Ödeme Onayı Mesaj Örneği

### Asp.Net

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<VPosMessageContract                               xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <OkUrl>http://localhost:10398/ThreeDModel/Approval</OkUrl>
    <FailUrl>http://localhost:10398/ThreeDModel/Fail</FailUrl>
    <HashData>DvAUXMvYV4ex5m16mMezEl+kxrl=</HashData>
    <MerchantId>1</MerchantId>
    <SubMerchantId>0</SubMerchantId>
    <MerchantOrderId>1554891870</MerchantOrderId>
    <InstallmentCount>0</InstallmentCount>
    <Amount>111</Amount>
    <FECAMount>0</FECAMount>
    <AdditionalData>
        <AdditionalDataList>
            <VPosAdditionalData>
                <Key>MD</Key>
                <Data>vygnTBD4smBxAOIDsgbaOQ==</Data>
            </VPosAdditionalData>
        </AdditionalDataList>
    </AdditionalData>
</VPosMessageContract>
```

```
</AdditionalDataList>
</AdditionalData>
<PaymentType>1</PaymentType>
<DebtId>0</DebtId>
<SurchargeAmount>0</SurchargeAmount>
<SGKDebtAmount>0</SGKDebtAmount>
<TranDate xsi:nil="true"/>
<InstallmentMaturityCommisionFlag>0</InstallmentMaturityCommisionFlag>
<CardHolderCustomerId xsi:nil="true"/>
<TransactionSecurity>3</TransactionSecurity>
<\VPosMessageContract>
```

## Ödeme Onay Sonucu Mesaj Yapısı

### Asp.Net

```
<?xml version="1.0" encoding="UTF-8"?>
<VPosTransactionResponseContract           xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    < VPosMessageContract >
        <OkUrl>http://localhost/ThreeDModel/Approval</OkUrl>
        <FailUrl>http://localhost/ThreeDModel/Fail</FailUrl>
        <HashData>DvAUXMvYV4ex5m16mMezEl+kxrl=</HashData>
        <MerchantId>1</MerchantId>
        <SubMerchantId>0</SubMerchantId>
        <CustomerId>936</CustomerId>
        <UserName>APIUSER</UserName>
        <HashPassword>kfkdsnskslkclswr9430ır</ HashPassword >
        <MerchantOrderId>1554891870</MerchantOrderId>
        <InstallmentCount>0</InstallmentCount>
        <Amount>111</Amount>
        <FECAMount>0</FECAMount>
        <AdditionalData>
            <AdditionalDataList>
```

```
<VPosAdditionalData>
    <Key>MD</Key>
    <Data>vygnTBD4smBxAOIDsgbaOQ==</Data>
</VPosAdditionalData>
</AdditionalDataList>
</AdditionalData>
<Products/>
<Addresses/>
<PaymentType>1</PaymentType>
<DebtId>0</DebtId>
<SurchargeAmount>0</SurchargeAmount>
<SGKDebtAmount>0</SGKDebtAmount>
<InstallmentMaturityCommisionFlag>0</InstallmentMaturityCommisionFlag>
<TransactionSecurity>3</TransactionSecurity>
</ VPosMessageContract >
<IsEnrolled>true</IsEnrolled>
<IsVirtual>false</IsVirtual>
<RRN>922709016599</RRN>
<Stan>016599</Stan>
<ResponseCode>00</ResponseCode>
<ResponseMessage>Provizyon Alindi.</ResponseMessage>
<OrderId>15161</OrderId>
<TransactionTime>00010101T00:00:00</TransactionTime>
<MerchantOrderId>1554891870</MerchantOrderId>
<HashData>bcCqBe4hbEIPOVYtfvsw7M44usQ=</HashData>
</VPosTransactionResponseContract>
```

## 3DSecure Ödeme İşlemi Örnek Kodlar

### Ödeme İşlemi Yapılması

#### Asp.Net

```
[HttpPost]
public VPosTransactionResponseContractPayment()
```

```
{
    VPosMessageContract request = new VPosMessageContract();
    request.PaymentType = 1;
    request.CardNumber = "";
    request.CardExpireDateMonth = "";
    request.CardExpireDateYear = "";
    request.CardCVV2 = "";
    request.CardHolderName = "";
    request.FECCurrencyCode = "";
    request.CurrencyCode = "0949";
    request.APIVersion = "";
    request.MerchantId = MerchantId;
    request.CustomerId = CustomerId;
    request.UserName = "";
    request.MerchantOrderId = "";
    request.InstallmentCount = 0;
    request.AdditionalData = new VPosAdditionalDataSet
    {
        AdditionalDataList = new List<VPosAdditionalData> {
            new VPosAdditionalData {
                Data = "asadasdsad",
                Description = "descdsadasd",
                Key = "keyasdadsads"
            }
        }
    };
    request.Amount = 1;
    request.DisplayAmount = "100";
    request.Addresses = new List<VPosAddressContract>();
    VPosAddressContract adres = new VPosAddressContract();
    adres.Name = "isminiz";
    adres.PhoneNumber = "telefon numaranız";
    adres.Email = "mail adresiniz";
    adres.AddressId = 1;
    adres.Type = "";
    request.Addresses.Add(adres);
    request.TransactionSecurity = 3;

    //HashData Hesaplanması
    string hashPassword = "";
    string hashString = CreateHashString(merchantId: request.MerchantId, merchantOrderId: "", amount: default(int), okUrl: "", failUrl: "", username: request.UserName, hashPassword: hashPassword);
    request.HashData = ComputeHash(hashString);

    VPosTransactionResponseContract response = new VPosTransactionResponseContract();

    XmlSerializer x = new XmlSerializer(request.GetType());
}
```

```

StringWriter sw = new StringWriter();
x.Serialize(sw, request);
XsdDataContractExporter exporter = new XsdDataContractExporter();
string requestMessage = sw.ToString();

string result =
DataPost("https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/ThreeDModelPayGate",
requestMessage);
try
{
    response =
DeserializeObject<VPosTransactionResponseContract>(result.Replace("\u00a0", "").Replace(
encoding=utf-16\"", ""));
}
catch (Exception ex)
{
    string errorMessage = "Mesaj1: " + ex.Message + ": Mesaj2: " + ex.StackTrace + ":" +
Mesaj3: " + ex.InnerException;
}

return response;
}

```

## İşlem Sonucunun Alınması

### Asp.Net

```

private VPosTransactionResponseContract GetPayloadResponse()
{
    string requestContent = string.Empty;
    requestContent = Request.Form["ResponseMessage"];
    requestContent = System.Web.HttpUtility.UrlDecode(requestContent);
    VPosTransactionResponseContract model =
        DeserializeObject<VPosTransactionResponseContract>(requestContent);
    return model;
}

```

## Ödeme İşleminin Onaylanması

### Asp.Net

```

[HttpPost]
public VPosTransactionResponseContract Approve(VPosTransactionResponseContract model)

{
    VPosMessageContract request = new VPosMessageContract();
    request.MerchantId = MerchantId;
    request.CustomerId = CustomerId;
    request.UserName = UserName;
}

```

```

request.TransactionSecurity = 3;
request.MerchantOrderId = "";
request.Amount = 111;
request.OkUrl = "https://domainname/ThreeDModel/Approval";
request.FailUrl = "http:// domainname /ThreeDModel/Fail";

request.AdditionalData = new VPosAdditionalDataSet();
request.AdditionalData.AdditionalDataList = new List<VPosAdditionalData>();
VPosAdditionalData md = new VPosAdditionalData();
md.Key = "MD";
md.Data = model.MD;
request.AdditionalData.AdditionalDataList.Add(md);

//HashData Hesaplanması
    string hashPassword = "";
    string hashString = CreateHashString(merchantId: request.MerchantId, merchantOrderId:
"", amount: default(int), okUrl: "", failUrl: "", username: request.UserName,
hashPassword: hashPassword);
    request.HashData = ComputeHash(hashString);

VPosTransactionResponseContract response = new VPosTransactionResponseContract();

XmlSerializer x = new XmlSerializer(request.GetType());
StringWriter sw = new StringWriter();
x.Serialize(sw, request);
XsdDataContractExporter exporter = new XsdDataContractExporter();
string requestMessage = sw.ToString();

string result =
DataPost("https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/ThreeDModelProvisionGate
", requestMessage);
try
{
    response =
DeserializeObject<VPosTransactionResponseContract>(result.Replace("\r\n", "").Replace(
encoding=utf-16", ""));
}
catch (Exception ex)
{
    string errorMessage = "Mesaj1: " + ex.Message + ": Mesaj2: " + ex.StackTrace + ":" +
Mesaj3: " + ex.InnerException;
}

return response;
}

```

### 3.1.3. Mail Order İşlemi

#### Mail Order İşlemi Mesaj Yapısı

<?xml version="1.0" encoding="UTF-16"?>

```
<VPosMessageContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<HashData> </HashData>
<MerchantId></MerchantId>
<SubMerchantId></SubMerchantId>
<CustomerId></CustomerId>
<UserName> </UserName>
<CustomerIPAddress></CustomerIPAddress>
<MerchantOrderId></MerchantOrderId>
<InstallmentCount></InstallmentCount>
<Amount></Amount>
<DisplayAmount></DisplayAmount>
<FECAmount></FECAmount>
<FECCurrencyCode></FECCurrencyCode> <Addresses>
    <VPosAddressContract>
        <Type></Type>
        <Name> </Name>
        <PhoneNumber></PhoneNumber>
        <OrderId></OrderId>
        <AddressId></AddressId>
        <Email> </Email>
    </VPosAddressContract>
</Addresses>
<APIVersion></APIVersion>
<CardNumber></CardNumber>
<CardExpireDateYear></CardExpireDateYear>
<CardExpireDateMonth></CardExpireDateMonth>
<CardCVV2></CardCVV2>
<CardHolderName> </CardHolderName>
<SurchargeAmount></SurchargeAmount>
```

```
<SGKDebtAmount></SGKDebtAmount>
<InstallmentMaturityCommisionFlag></InstallmentMaturityCommisionFlag>
<TransactionSecurity></TransactionSecurity>
<CardGuid></CardGuid>
</VPosMessageContract>
```

### Mail Order İşlemi Mesaj Örneği

```
<?xml version="1.0" encoding="UTF-16"?>

<VPosMessageContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<HashData>vCZmz5eixuSt/HTDEcL/8bt8ZAE=</HashData>
<MerchantId>1</MerchantId>
<SubMerchantId>0</SubMerchantId>
<CustomerId>11111</CustomerId>
<UserName>APIUSER</UserName>
<HashPassword>kfkdsnskslkclswr9430ır</ HashPassword >
<MerchantOrderId>2126497214</MerchantOrderId>
<InstallmentCount>0</InstallmentCount>
<Amount>120</Amount>
<DisplayAmount>120</DisplayAmount>
<FECAMount>0</FECAMount>
<FECCurrencyCode>0949</FECCurrencyCode> <Addresses>
    <VPosAddressContract>
        <Type>1</Type>
        <Name>Mahmut Sami YAZAR</Name>
        <PhoneNumber>324234234234</PhoneNumber>
        <OrderId>0</OrderId>
        <AddressId>12</AddressId>
        <Email>mahmutsamiyazar@hotmail.com</Email>
```

```
</VPosAddressContract>

</Addresses>

<APIVersion>1.0.0</APIVersion>

<CardNumber>5353550000958906</CardNumber>

<CardExpireDateYear>23</CardExpireDateYear>

<CardExpireDateMonth>01</CardExpireDateMonth>

<CardCVV2>741</CardCVV2>

<CardHolderName>Hasan Karacan</CardHolderName>

<DebtId>0</DebtId>

<SurchargeAmount>0</SurchargeAmount>

<SGKDebtAmount>0</SGKDebtAmount>

<InstallmentMaturityCommisionFlag>0</InstallmentMaturityCommisionFlag>

<TransactionSecurity>1</TransactionSecurity>

<CardGuid>AA9588EF350C480FBE5CAD40A463AF00</CardGuid>

</VPosMessageContract>
```

### Mail Order İşlemi Sonucu Mesaj Yapısı

```
<?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>

  <HashData> </HashData>

  <MerchantId></MerchantId>

  <SubMerchantId></SubMerchantId>

  <CustomerId></CustomerId>

  <UserName> </UserName>

  <CustomerIPAddress></CustomerIPAddress>

  <BatchId></BatchId>

  <MerchantOrderId></MerchantOrderId>

  <InstallmentCount></InstallmentCount>

  <Amount></Amount>
```

```
<DisplayAmount></DisplayAmount>
<FECAmount></FECAmount>
<FECCurrencyCode></FECCurrencyCode>
<Products/
<Addresses>
    <VPosAddressContract>
        <Type></Type>
        <Name> </Name>
        <PhoneNumber></PhoneNumber>
        <OrderId></OrderId>
        <AddressId></AddressId>
        <Email> </Email>
    </VPosAddressContract>
</Addresses>
<APIVersion></APIVersion>
<CardNumber></CardNumber>
<CardHolderName> </CardHolderName>
<SurchargeAmount></SurchargeAmount>
<SGKDebtAmount></SGKDebtAmount>
<InstallmentMaturityCommisionFlag></InstallmentMaturityCommisionFlag>
<TransactionSecurity></TransactionSecurity>
<CardGuide></CardGuide>
</VPosMessage>
<IsEnrolled> </IsEnrolled>
<IsVirtual> </IsVirtual>
<RRN></RRN>
<Stan></Stan>
<ResponseCode></ResponseCode>
<ResponseMessage></ResponseMessage>
<OrderId></OrderId>
```

```
<TransactionTime></TransactionTime>  
<MerchantOrderId></MerchantOrderId>  
<HashData></HashData>  
<BusinessKey></BusinessKey>  
</VPosTransactionResponseContract>
```

## Mail Order İşlemi Örnek Kodlar

### Mail Order İşleminin Yapılması

#### Asp.Net

```
[HttpPost]  
public VPosTransactionResponseContract MailOrder()  
{  
    VPosMessageContract request = new VPosMessageContract();  
    request.PaymentType = 1;  
    request.CardNumber = "";  
    request.CardExpireDateMonth = "";  
    request.CardExpireDateYear = "";  
    request.CardCVV2 = "";  
    request.CardHolderName = "";  
    request.FECCurrencyCode = "";  
    request.CurrencyCode = "0949";  
    request.APIVersion = "";  
    request.MerchantId = MerchantId;  
    request.CustomerId = CustomerId;  
    request.UserName = "";  
    request.MerchantOrderId = "";  
    request.InstallmentCount = "";  
    request.Amount = 1;  
    request.DisplayAmount = "100";  
    request.Addresses = new List<VPosAddressContract>();  
    VPosAddressContract adres = new VPosAddressContract();  
    adres.Name = "isminiz";  
    adres.PhoneNumber = "telefon numaranız";  
    adres.Email = "mail adresiniz";  
    adres.AddressId = ;  
    adres.Type = "";  
    request.Addresses.Add(adres);  
    request.TransactionSecurity = 1;  
    request.CardGuid = "AA9588EF350C480FBE5CAD401233AF00"; //eğer kaydedilmiş kart ile işlemse bu alan doldu  
    gönderilmeli  
  
    //HashData Hesaplanması
```

```

        string hashPassword = "";
        string hashString = CreateHashString(merchantId: request.MerchantId, merchantOrderId:
        "", amount: default(int), okUrl: "", failUrl: "", username: request.UserName,
        hashPassword: hashPassword);
        request.HashData = ComputeHash(hashString);

        VPosTransactionResponseContract response = new VPosTransactionResponseContract();

        XmlSerializer x = new XmlSerializer(request.GetType());
        StringWriter sw = new StringWriter();
        x.Serialize(sw, request);
        XsdDataContractExporter exporter = new XsdDataContractExporter();
        string requestMessage = sw.ToString();

        string result =
DataPost("https://boa.vakifkatlim.com.tr/VirtualPOS.Gateway/Home/MailOrderSale",
requestMessage);
        try
        {
            response =
DeserialzeObject<VPosTransactionResponseContract>(result.Replace("\u00a0", "").Replace(
encoding="utf-16\"", ""));
        }
        catch (Exception ex)
        {
            string errorMessage = "Mesaj1: " + ex.Message + ": Mesaj2: " + ex.StackTrace + ":" +
Mesaj3: " + ex.InnerException;
        }

        return response;
    }
}

```

### 3.1.4. Sigorta Şirketleri Mail Order İşlemi

#### Sigorta Şirketleri Mail Order İşlemi Mesaj Yapısı

```

<?xml version="1.0" encoding="utf-16"?>
<VPosMessageContract xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<HashData></HashData>
<MerchantId></MerchantId>
<SubMerchantId></SubMerchantId>
<CustomerId></CustomerId>
<UserName></UserName>
<CustomerIPAddress></CustomerIPAddress>
<MerchantOrderId></MerchantOrderId>
<InstallmentCount></InstallmentCount>
<Amount></Amount>
<DisplayAmount></DisplayAmount>
<FECCurrencyCode></FECCurrencyCode>

```

```
<TransactionSecurity></TransactionSecurity>
<Addresses>
  <VPosAddressContract>
    <Type></Type>
    <Name></Name>
    <PhoneNumber></PhoneNumber>
    <OrderId></OrderId>
    <AddressId></AddressId>
    <Email></Email>
  </VPosAddressContract>
</Addresses>
<CardNumber></CardNumber>
<CardHolderName></CardHolderName>
<VPSEntryMode></VPSEntryMode>
<IdentityTaxNumber></IdentityTaxNumber>
<CardGuid></CardGuid>
</VPosMessageContract>
```

## Sigorta Şirketleri Mail Order İşleme Mesaj Örneği

```
<?xml version="1.0" encoding="utf-16"?>
<VPosMessageContract xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<HashData>4fxaa6XGHp4hldQMg5hsI59hg=</HashData>
<MerchantId>1</MerchantId>
<SubMerchantId>0</SubMerchantId>
<CustomerId>11111</CustomerId>
<UserName>USER</UserName>
<CustomerIPAddress>1</CustomerIPAddress>
<MerchantOrderId>194837123</MerchantOrderId>
<InstallmentCount>0</InstallmentCount>
<Amount>350</Amount>
<DisplayAmount>350</DisplayAmount>
<FECCurrencyCode>0949</FECCurrencyCode>
<TransactionSecurity>0</TransactionSecurity>
<Addresses>
  <VPosAddressContract>
    <Type>1</Type>
    <Name>Mahmut Yılmaz</Name>
    <PhoneNumber>324234234234</PhoneNumber>
    <OrderId>0</OrderId>
    <AddressId>12</AddressId>
    <Email>mahmutyilmaz</Email>
  </VPosAddressContract>
</Addresses>
<APIVersion>1.0.0</APIVersion>
<CardNumber>547980*6543</CardNumber>
<CardHolderName>Hasan Karacan</CardHolderName>
<CardType>MasterCard</CardType>
<CardGuid>AA9588EF350C480FB5CAD40A463AF00</CardGuid>
<IdentityTaxNumber>10070186788</IdentityTaxNumber>
```

</VPosMessageContract>

## Sigorta Şirketleri Mail Order İşlemi Sonucu Mesaj Yapısı

```
<?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>

    <HashData> </HashData>

    <MerchantId></MerchantId>

    <SubMerchantId></SubMerchantId>

    <CustomerId></CustomerId>

    <UserName> </UserName>

    <CustomerIPAddress></CustomerIPAddress>

    <BatchId></BatchId>

    <MerchantOrderId></MerchantOrderId>

    <InstallmentCount></InstallmentCount>

    <Amount></Amount>

    <DisplayAmount></DisplayAmount>

    <FECAMount></FECAMount>

    <FECCurrencyCode></FECCurrencyCode>

    <Products/>

    <Addresses>

        <VPosAddressContract>

            <Type></Type>

            <Name> </Name>

            <PhoneNumber></PhoneNumber>

            <OrderId></OrderId>

            <AddressId></AddressId>

            <Email> </Email>

        </VPosAddressContract>

    </Addresses>

    <APIVersion></APIVersion>
```

```
<CardNumber></CardNumber>
<CardHolderName> </CardHolderName>
<SurchargeAmount></SurchargeAmount>
<SGKDebtAmount></SGKDebtAmount>
<InstallmentMaturityCommisionFlag></InstallmentMaturityCommisionFlag>
<TransactionSecurity></TransactionSecurity>

</VPosMessage>
<IsEnrolled> </IsEnrolled>
<IsVirtual> </IsVirtual>
<RRN></RRN>
<Stan></Stan>
<ResponseCode></ResponseCode>
<ResponseMessage></ResponseMessage>
<OrderId></OrderId>
<TransactionTime></TransactionTime>
<MerchantOrderId></MerchantOrderId>
<HashData></HashData>
<BusinessKey></BusinessKey>
</VPosTransactionResponseContract>
```

## Sigorta Şirketleri Mail Order İşlemi Örnek Kodlar

### Sigorta Şirketleri Mail Order İşleminin Yapılması

#### Asp.Net

```
[HttpPost]
public VPosTransactionResponseContract MailOrder()
{
    VPosMessageContract request = new VPosMessageContract();
    request.PaymentType = 1;
    request.CardNumber = "123456*1234";
    request.IdentityTaxNumber = "11223344556";
    request.CardHolderName = "";
    request.FECCurrencyCode = "";
    request.CurrencyCode = "0949";
    request.APIVersion = "";
```

```

request.MerchantId = MerchantId;
request.CustomerId = CustomerId;
request.UserName = "";
request.MerchantOrderId = "";
request.InstallmentCount = "";
request.Amount = 1;
request.DisplayAmount = "100";
request.Addresses = new List<VPosAddressContract>();
VPosAddressContract adres = new VPosAddressContract();
adres.Name = "isminiz";
adres.PhoneNumber = "telefon numaranız";
adres.Email = "mail adresiniz";
adres.AddressId = 1 ;
adres.Type = "";
request.Addresses.Add(adres);
request.TransactionSecurity = 1;
request.CardGuid = "AA9588EF350C480FBE5CAD401233AF00"; //eğer kaydedilmiş kart ile işlemse bu alan dolu
gönderilmeli

```

```

//HashData Hesaplanması

    string hashPassword = "";
    string hashString = CreateHashString(merchantId: request.MerchantId, merchantOrderId:
 "", amount: default(int), okUrl: "", failUrl: "", username: request.UserName,
hashPassword: hashPassword);
    request.HashData = ComputeHash(hashString);

    VPosTransactionResponseContract response = new VPosTransactionResponseContract();

    XmlSerializer x = new XmlSerializer(request.GetType());
    StringWriter sw = new StringWriter();
    x.Serialize(sw, request);
    XsdDataContractExporter exporter = new XsdDataContractExporter();
    string requestMessage = sw.ToString();

    string result =
DataPost("https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/MailOrderSale",
requestMessage);
    try
    {
        response =
Deserializable<VPosTransactionResponseContract>(result.Replace("\r\n", "").Replace(
encoding="utf-16\"", ""));
    }
    catch (Exception ex)
    {
        string errorMessage = "Mesaj1: " + ex.Message + ": Mesaj2: " + ex.StackTrace + ":" +
Mesaj3: " + ex.InnerException;
    }

```

```
    return response;  
}
```

### 3.1.5. İptal İşlemi

#### Satış İptali Mesaj Yapısı

```
<?xml version="1.0" encoding="UTF-16"?>  
<VPosMessageContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
    <HashData></HashData>  
    <MerchantId></MerchantId>  
    <SubMerchantId></SubMerchantId>  
    <CustomerId></CustomerId>  
    <UserName></UserName>  
    <MerchantOrderId></MerchantOrderId>  
    <Amount></Amount>  
    <OrderId></OrderId>  
    <PaymentType> </PaymentType>  
</VPosMessageContract>
```

#### Satış İptali Mesaj Örneği

```
<?xml version="1.0" encoding="UTF-16"?>  
<VPosMessageContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
    <HashData>ApVe78JRHef7ukolY5il4p0J9ok=</HashData>  
    <MerchantId>1</MerchantId>  
    <SubMerchantId>0</SubMerchantId>  
    <CustomerId>936</CustomerId>  
    <UserName>APIUSER</UserName> <HashPassword>kfkdsnskslkclswr9430ır</  
HashPassword>  
    <MerchantOrderId>1061162073</MerchantOrderId>  
    <Amount>970</Amount>  
    <OrderId>12345</OrderId>  
    <PaymentType>1</PaymentType>  
</VPosMessageContract>
```

## Satış İptali Sonucu Mesaj Yapısı

```
<?xml version="1.0" encoding="UTF-8"?>
<VPosTransactionResponseContract
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <VPosMessage>
  <HashData></HashData>
  <MerchantId></MerchantId>
  <SubMerchantId></SubMerchantId>
  <CustomerId></CustomerId>
  <UserName> </UserName>
  <CustomerIPAddress></CustomerIPAddress>
  <BatchId></BatchId>
  <MerchantOrderId></MerchantOrderId>
  <InstallmentCount></InstallmentCount>
  <Amount></Amount>
  <DisplayAmount></DisplayAmount>
  <FECAmount></FECAmount>
  <FECCurrencyCode></FECCurrencyCode>
  <Addresses>
    <VPosAddressContract>
      <Type></Type>
      <Name></Name>
      <PhoneNumber></PhoneNumber>
      <OrderId></OrderId>
      <AddressId></AddressId>
      <Email></Email>
    </VPosAddressContract>
  </Addresses>
  <APIVersion></APIVersion>
  <PaymentType> </PaymentType>
```

```
<SurchargeAmount></SurchargeAmount>
<SGKDebtAmount></SGKDebtAmount>
<InstallmentMaturityCommisionFlag></InstallmentMaturityCommisionFlag>
<TransactionSecurity></TransactionSecurity>
</VPosMessage>
<IsEnrolled> </IsEnrolled>
<IsVirtual> </IsVirtual>
<ResponseCode> </ResponseCode>
<ResponseMessage> </ResponseMessage>
<OrderId></OrderId>
<TransactionTime></TransactionTime>
<BusinessKey></BusinessKey>
</VPosTransactionResponseContract>
```

## İptal İşlemi Örnek Kodlar

### Satış İptali Yapılması

#### Asp.Net

```
[HttpPost]
public VPosTransactionResponseContract SaleReversal()
{
    VPosMessageContract request = new VPosMessageContract();
    request.PaymentType = 1;
    request.CardNumber = "";
    request.CardExpireDateMonth = "";
    request.CardExpireDateYear = "";
    request.CardCVV2 = "";
    request.CardHolderName = "";
    request.FECCurrencyCode = "";
    request.CurrencyCode = "0949";
    request.APIVersion = "";
    request.MerchantId = MerchantId;
    request.CustomerId = CustomerId;
    request.UserName = "";
    request.MerchantOrderId = "";
    request.InstallmentCount = "";
    request.Amount = 1;
    request.DisplayAmount = "100";
    request.Addresses = new List<VPosAddressContract>();
```

```

VPosAddressContract adres = new VPosAddressContract();
adres.Name = "isminiz";
adres.PhoneNumber = "telefon numaranız";
adres.Email = "mail adresiniz";
adres.AddressId = ;
adres.Type = "";
request.Addresses.Add(adres);

//HashData Hesaplanması

    string hashPassword = "";
    string hashString = CreateHashString(merchantId: request.MerchantId, merchantOrderId:
    "", amount: default(int), okUrl: "", failUrl: "", username: request.UserName,
hashPassword: hashPassword);
    request.HashData = ComputeHash(hashString);

    VPosTransactionResponseContract response = new VPosTransactionResponseContract();

    XmlSerializer x = new XmlSerializer(request.GetType());
    StringWriter sw = new StringWriter();
    x.Serialize(sw, request);
    XsdDataContractExporter exporter = new XsdDataContractExporter();
    string requestMessage = sw.ToString();

    string result =
DataPost("https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/SaleReversal",
requestMessage);
    try
    {
        response =
DeserializeObject<VPosTransactionResponseContract>(result.Replace("\r\n", "").Replace(
"encoding=utf-16\"", ""));
    }
    catch (Exception ex)
    {
        string errorMessage = "Mesaj1: " + ex.Message + ": Mesaj2: " + ex.StackTrace + ":" +
Mesaj3: " + ex.InnerException;
    }

    return response;
}

```

### 3.1.6. İade İşlemi

#### İade İşlemi Mesaj Yapısı

```

<?xml version="1.0" encoding="UTF-16"?>
<VPosMessageContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<HashData> </HashData>

```

```
<MerchantId></MerchantId>
<SubMerchantId></SubMerchantId>
<CustomerId></CustomerId>
<UserName> </UserName>
<CustomerIPAddress></CustomerIPAddress>
<OrderId></OrderId>
<MerchantOrderId></MerchantOrderId>
</VPosMessageContract>
```

### **İade İşlemi Mesaj Örneği**

```
<?xml version="1.0" encoding="UTF-16"?>
<VPosMessageContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <HashData>t6XJZeNvT+wKtj4GsHZvodG9y3E=</HashData>
    <MerchantId>1</MerchantId>
    <SubMerchantId>0</SubMerchantId>
    <CustomerId>11111</CustomerId>
    <UserName>APIUSER</UserName>
    <HashPassword>kfkdsnskslkclswr9430ır</ HashPassword >
    <OrderId>15184</OrderId>
    <MerchantOrderId>1771489024</MerchantOrderId>
</VPosMessageContract>
```

### **İade İşlemi Sonucu Mesaj Yapısı**

```
<?xml version="1.0" encoding="UTF-8"?>
<VPosTransactionResponseContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <VPosMessage>
        <HashData> </HashData>
        <MerchantId></MerchantId>
        <SubMerchantId></SubMerchantId>
        <CustomerId></CustomerId>
        <UserName> </UserName>
```

```
<CustomerIPAddress></CustomerIPAddress>
<OrderId></OrderId>
<BatchId></BatchId>
<MerchantOrderId></MerchantOrderId>
<InstallmentCount></InstallmentCount>
<Amount></Amount>
<DisplayAmount></DisplayAmount>
<FECAmount></FECAmount>
<FECCurrencyCode></FECCurrencyCode>
<Products/>
<Addresses>
    <VPosAddressContract>
        <Type></Type>
        <Name> </Name>
        <PhoneNumber></PhoneNumber>
        <OrderId></OrderId>
        <AddressId></AddressId>
        <Email> </Email>
    </VPosAddressContract>
</Addresses>
<APIVersion></APIVersion>
<PaymentType> </PaymentType>
<SurchargeAmount></SurchargeAmount>
<SGKDebtAmount></SGKDebtAmount>
<InstallmentMaturityCommisionFlag></InstallmentMaturityCommisionFlag>
<TransactionSecurity></TransactionSecurity>
</VPosMessage>
<IsEnrolled> </IsEnrolled>
<IsVirtual> </IsVirtual>
<RRN></RRN>
```

```
<Stan></Stan>

<ResponseCode></ResponseCode>

<ResponseMessage> </ResponseMessage>

<OrderId></OrderId>

<TransactionTime></TransactionTime>

<MerchantOrderId></MerchantOrderId>

<BusinessKey></BusinessKey>

</VPosTransactionResponseContract>
```

## İade İşlemi Örnek Kodlar

### İade İşlemi Yapılması

#### Asp.Net

```
[HttpPost]
public VPosTransactionResponseContract DrawBack()
{
    VPosMessageContract request = new VPosMessageContract();
    request.MerchantId = MerchantId;
    request.CustomerId = CustomerId;
    request.UserName UserName;
    request.MerchantOrderId = MerchantOrderId;
    request.OrderId = OrderId;

    //HashData Hesaplanması

    string hashPassword = "";
    string hashString = CreateHashString(merchantId: request.MerchantId, merchantOrderId:
    "", amount: default(int), okUrl: "", failUrl: "", username: request.UserName,
    hashPassword: hashPassword);
    request.HashData = ComputeHash(hashString);

    VPosTransactionResponseContract response = new VPosTransactionResponseContract();

    XmlSerializer x = new XmlSerializer(request.GetType());
    StringWriter sw = new StringWriter();
    x.Serialize(sw, request);
    XsdDataContractExporter exporter = new XsdDataContractExporter();
    string requestMessage = sw.ToString();

    string result =
    DataPost("https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/DrawBack",
    requestMessage);
    try
    {
```

```
        response =
DeserializeObject<VPosTransactionResponseContract>(result.Replace("\u0000", "").Replace(
encoding="utf-16\"", ""));
    }
    catch (Exception ex)
{
    string errorMessage = "Mesaj1: " + ex.Message + ": Mesaj2: " + ex.StackTrace + ":" +
Mesaj3: " + ex.InnerException;
}

return response;
}
```

### 3.1.7. Kısmi İade İşlemi

#### Kısmi İade İşlemi Mesaj Yapısı

```
<?xml version="1.0" encoding="UTF-16"?>
<VPosMessageContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <HashData> </HashData>
    <MerchantId></MerchantId>
    <SubMerchantId></SubMerchantId>
    <CustomerId></CustomerId>
    <UserName> </UserName>
    <CustomerIPAddress></CustomerIPAddress>
    <OrderId></OrderId>
    <MerchantOrderId></MerchantOrderId>
    <Amount></Amount>
    <DisplayAmount></DisplayAmount>
</VPosMessageContract>
```

#### Kısmi İade İşlemi Mesaj Örneği

```
<?xml version="1.0" encoding="UTF-16"?>
<VPosMessageContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <HashData>t6XJZeNvT+wKtj4GsHZvodG9y3E=</HashData>
    <MerchantId>1</MerchantId>
    <SubMerchantId>0</SubMerchantId>
    <CustomerId>11111</CustomerId>
```

```
<UserName>APIUSER</UserName>
<HashPassword>kfkdsnskslkclswr9430ır</ HashPassword >
<OrderId>15184</OrderId>
<MerchantOrderId>1771489024</MerchantOrderId>
<Amount>500</Amount>
<DisplayAmount>500</DisplayAmount>
</VPosMessageContract>
```

### Kısmi İade İşlemi Sonucu Mesaj Yapısı

```
<?xml version="1.0" encoding="UTF-8"?>
<VPosTransactionResponseContract
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <VPosMessage>
  <HashData> HashData</HashData>
  <MerchantId></MerchantId>
  <SubMerchantId></SubMerchantId>
  <CustomerId></CustomerId>
  <UserName> </UserName>
  <CustomerIPAddress></CustomerIPAddress>
  <OrderId></OrderId>
  <BatchId></BatchId>
  <MerchantOrderId></MerchantOrderId>
  <InstallmentCount></InstallmentCount>
  <Amount></Amount>
  <DisplayAmount></DisplayAmount>
  <FECAmount></FECAmount>
  <FECCurrencyCode></FECCurrencyCode>
  <Products/>
  <Addresses>
    <VPosAddressContract>
      <Type></Type>
      <Name> </Name>
```

```
<PhoneNumber></PhoneNumber>
<OrderId></OrderId>
<AddressId></AddressId>
<Email> </Email>
</VPosAddressContract>

</Addresses>
<APIVersion></APIVersion>
<PaymentType> </PaymentType>
<SurchargeAmount></SurchargeAmount>
<SGKDebtAmount></SGKDebtAmount>
<InstallmentMaturityCommisionFlag></InstallmentMaturityCommisionFlag>
<TransactionSecurity></TransactionSecurity>

</VPosMessage>
<IsEnrolled> </IsEnrolled>
<IsVirtual> </IsVirtual>
<RRN></RRN>
<Stan></Stan>
<ResponseCode></ResponseCode>
<ResponseMessage></ResponseMessage>
<OrderId></OrderId>
<TransactionTime></TransactionTime>
<MerchantOrderId></MerchantOrderId>
<BusinessKey></BusinessKey>
</VPosTransactionResponseContract>
```

## Kısmi İade İşlemi Örnek Kodlar

### Kısmi İade İşlemi Yapılması

#### Asp.Net

```
[HttpPost]
public VPosTransactionResponseContract PartialDrawBack()
```

```
{
    VPosMessageContract request = new VPosMessageContract();
    request.MerchantId = MerchantId;
    request.CustomerId = CustomerId;
    request.UserName = UserName;
    request.MerchantOrderId = MerchantOrderId;
    request.Amount = Amount;
    request.DisplayAmount = request.Amount.ToString();
    request.OrderId = OrderId;

    //HashData Hesaplanması

    string hashPassword = "";
    string hashString = CreateHashString(merchantId: request.MerchantId, merchantOrderId:
    "", amount: default(int), okUrl: "", failUrl: "", username: request.UserName,
    hashPassword: hashPassword);
    request.HashData = ComputeHash(hashString);

    VPosTransactionResponseContract response = new VPosTransactionResponseContract();

    XmlSerializer x = new XmlSerializer(request.GetType());
    StringWriter sw = new StringWriter();
    x.Serialize(sw, request);
    XsdDataContractExporter exporter = new XsdDataContractExporter();
    string requestMessage = sw.ToString();

    string result =
    DataPost("https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/PartialDrawBack",
    requestMessage);
    try
    {
        response =
        DeserializeObject<VPosTransactionResponseContract>(result.Replace("\u0000", "").Replace(
        encoding=utf-16", ""));
    }
    catch (Exception ex)
    {
        string errorMessage = "Mesaj1: " + ex.Message + ": Mesaj2: " + ex.StackTrace + ":" +
        Mesaj3: " + ex.InnerException;
    }

    return response;
}
```

### 3.1.8. Ön Provizyon İşlemi

#### Ön Provizyon İşlemi Mesaj Yapısı

```
<?xml version="1.0" encoding="UTF-16"?>
```

```
<VPosMessageContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <HashData> </HashData>
    <MerchantId></MerchantId>
    <SubMerchantId></SubMerchantId>
    <CustomerId></CustomerId>
    <UserName> </UserName>
    <CustomerIPAddress></CustomerIPAddress>
    <BatchId></BatchId>
    <MerchantOrderId></MerchantOrderId>
    <InstallmentCount></InstallmentCount>
    <Amount></Amount>
    <DisplayAmount></DisplayAmount>
    <FECAmount></FECAmount>
    <FECCurrencyCode></FECCurrencyCode> <Addresses
        <VPosAddressContract>
            <Type></Type>
            <Name> </Name>
            <PhoneNumber></PhoneNumber>
            <OrderId></OrderId>
            <AddressId></AddressId>
            <Email> </Email>
        </VPosAddressContract>
    </Addresses>
    <APIVersion></APIVersion>
    <CardNumber></CardNumber>
    <CardExpireDateYear></CardExpireDateYear>
    <CardExpireDateMonth></CardExpireDateMonth>
    <CardCVV2></CardCVV2>
    <CardHolderName> </CardHolderName>
    <PaymentType> </PaymentType>
    <SurchargeAmount></SurchargeAmount> <SGKDebtAmount></SGKDebtAmount>
    <InstallmentMaturityCommisionFlag></InstallmentMaturityCommisionFlag>
    <TransactionSecurity></TransactionSecurity>
</VPosMessageContract>
```

## Ön Provizyon İşlemi Mesaj Örneği

```
<?xml version="1.0" encoding="UTF-16"?>
<VPosMessageContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <HashData>fY5CK7nmMa9WkVdhV+u2Bp557n4=</HashData>
    <MerchantId>1</MerchantId>
    <SubMerchantId>0</SubMerchantId>
    <CustomerId>11111</CustomerId>
    <UserName>APIUSER</UserName>
    <HashPassword>kfkdsnskslkclswr9430ır</ HashPassword >
    <MerchantOrderId>58957265</MerchantOrderId>
    <InstallmentCount>0</InstallmentCount>
    <Amount>840</Amount>
    <DisplayAmount>840</DisplayAmount>
    <FECAMount>0</FECAMount>
    <FECCurrencyCode>0949</FECCurrencyCode <Addresses
        <VPosAddressContract>
            <Type>1</Type>
            <Name>Mahmut Sami YAZAR</Name>
            <PhoneNumber>324234234234</PhoneNumber>
            <OrderId>0</OrderId>
            <AddressId>12</AddressId>
            <Email>mahmutsamiyazar@hotmail.com</Email>
        </VPosAddressContract>
    </Addresses>
    <APIVersion>1.0.0</APIVersion>
    <CardNumber>5353550000958906</CardNumber>
    <CardExpireDateYear>23</CardExpireDateYear>
    <CardExpireDateMonth>01</CardExpireDateMonth>
    <CardCVV2>741</CardCVV2>
    <CardHolderName>Hasan Karacan</CardHolderName>
    <PaymentType>1</PaymentType>
    <DebtId>0</DebtId>
    <SurchargeAmount>0</SurchargeAmount>
```

```
<SGKDebtAmount>0</SGKDebtAmount>
<InstallmentMaturityCommisionFlag>0</InstallmentMaturityCommisionFlag>
<TransactionSecurity>5</TransactionSecurity>
</VPosMessageContract>
```

## Ön Provizyon İşlemi Sonucu Mesaj Yapısı

```
<?xml version="1.0" encoding="UTF-8"?>
<VPosTransactionResponseContract
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <VPosMessage>
  <HashData>fY5CK7nmMa9WkVdhV+u2Bp557n4=</HashData>
  <MerchantId>1</MerchantId>
  <SubMerchantId>0</SubMerchantId>
  <CustomerId>11111</CustomerId>
  <UserName>APIUSER</UserName>
  <HashPassword>kfkdsnskslkclswr9430ır</ HashPassword >
  <MerchantOrderId>58957265</MerchantOrderId>
  <InstallmentCount>0</InstallmentCount>
  <Amount>840</Amount>
  <DisplayAmount>840</DisplayAmount>
  <FECAMount>0</FECAMount>
  <FECCurrencyCode>0949</FECCurrencyCode>
  <Products/>
  <Addresses>
    <VPosAddressContract>
      <Type>1</Type>
      <Name>Mahmut Sami YAZAR</Name>
      <PhoneNumber>324234234234</PhoneNumber>
      <OrderId>0</OrderId>
      <AddressId>12</AddressId>
      <Email>mahmutsamiyazar@hotmail.com</Email>
    </VPosAddressContract>
  </Addresses>
</VPosMessage>
```

```
</VPosAddressContract>

</Addresses>

<APIVersion>1.0.0</APIVersion>

<CardNumber>5353550000958906</CardNumber>

<CardHolderName>Hasan Karacan</CardHolderName>

<PaymentType>None</PaymentType>

<DebtId>0</DebtId>

<SurchargeAmount>0</SurchargeAmount>

<SGKDebtAmount>0</SGKDebtAmount>

<InstallmentMaturityCommisionFlag>0</InstallmentMaturityCommisionFlag>

<TransactionSecurity>5</TransactionSecurity>

</VPosMessage>

<IsEnrolled>true</IsEnrolled>

<IsVirtual>false</IsVirtual>

<RRN>922810016639</RRN>

<Stan>016639</Stan>

<ResponseCode>51</ResponseCode>

<ResponseMessage>Limit Yetersiz.</ResponseMessage>

<OrderId>15188</OrderId>

<TransactionTime>2019-08-16T10:54:23.81069</TransactionTime>

<MerchantOrderId>58957265</MerchantOrderId>

<BusinessKey>0</BusinessKey>

</VPosTransactionResponseContract>
```

## Ön Provizyon İşlemi Örnek Kodlar

### Ön Provizyon İşlemi Yapılması

#### Asp.Net

```
[HttpPost]
public VPosTransactionResponseContract PreAuthorization()
{
```

```

VPosMessageContract request = new VPosMessageContract();
request.PaymentType = 1;
request.CardNumber = "";
request.CardExpireDateMonth = "";
request.CardExpireDateYear = "";
request.CardCVV2 = "";
request.CardHolderName = "";
request.FECCurrencyCode = "";
request.CurrencyCode = "0949";
request.APIVersion = "";
request.MerchantId = MerchantId;
request.CustomerId = CustomerId;
request.UserName = "";
request.MerchantOrderId = "";
request.InstallmentCount = "";
request.Amount = 1;
request.DisplayAmount = "100";
request.Addresses = new List<VPosAddressContract>();
VPosAddressContract adres = new VPosAddressContract();
adres.Name = "isminiz";
adres.PhoneNumber = "telefon numaranız";
adres.Email = "mail adresiniz";
adres.AddressId = ;
adres.Type = "";
request.Addresses.Add(adres);
request.TransactionSecurity = 5;

//HashData Hesaplanması

    string hashPassword = "";
    string hashString = CreateHashString(merchantId: request.MerchantId, merchantOrderId:
    "", amount: default(int), okUrl: "", failUrl: "", username: request.UserName,
    hashPassword: hashPassword);
    request.HashData = ComputeHash(hashString);

    VPosTransactionResponseContract response = new VPosTransactionResponseContract();

    XmlSerializer x = new XmlSerializer(request.GetType());
    StringWriter sw = new StringWriter();
    x.Serialize(sw, request);
    XsdDataContractExporter exporter = new XsdDataContractExporter();
    string requestMessage = sw.ToString();

    string result =
DataPost("https://boa.vakifikatilim.com.tr/VirtualPOS.Gateway/Home/PreAuthorizaten",
requestMessage);
    try
    {

```

```
response =
DeserializeObject<VPosTransactionResponseContract>(result.Replace("\u0000", "").Replace(
encoding="utf-16\"", ""));
}
catch (Exception ex)
{
    string errorMessage = "Mesaj1: " + ex.Message + ": Mesaj2: " + ex.StackTrace + ":" +
Mesaj3: " + ex.InnerException;
}

return response;
}
```

### 3.1.9. Ön Provizyon Kapama İşleme

#### Ön Provizyon Kapama İşleme Mesaj Yapısı

```
<?xml version="1.0" encoding="UTF-16"?>

<VPosMessageContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<HashData> </HashData>

<MerchantId></MerchantId>

<SubMerchantId></SubMerchantId>

<CustomerId></CustomerId>

<UserName> </UserName>

<CustomerIPAddress></CustomerIPAddress>

<OrderId></OrderId>

<MerchantOrderId></MerchantOrderId>

</VPosMessageContract>
```

#### Ön Provizyon Kapama İşleme Mesaj Örneği

```
<?xml version="1.0" encoding="UTF-16"?>

<VPosMessageContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<HashData>t6XJZeNvT+wKtj4GsHZvodG9y3E=</HashData>

<MerchantId>1</MerchantId>

<SubMerchantId>0</SubMerchantId>
```

```
<CustomerId>11111</CustomerId>
<UserName>APIUSER</UserName>
<HashPassword>kfkdsnskslkclswr9430ır</ HashPassword >
<OrderId>15184</OrderId>
<MerchantOrderId>1771489024</MerchantOrderId>
</VPosMessageContract>
```

### Ön Provizyon Kapama İşlemi Sonucu Mesaj Yapısı

```
<?xml version="1.0" encoding="UTF-8"?>
<VPosTransactionResponseContract
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <VPosMessage>
    <HashData> </HashData>
    <MerchantId></MerchantId>
    <SubMerchantId></SubMerchantId>
    <CustomerId></CustomerId>
    <UserName> </UserName>
    <CustomerIPAddress></CustomerIPAddress>
    <OrderId></OrderId>
    <MerchantOrderId></MerchantOrderId>
    <InstallmentCount></InstallmentCount>
    <Amount></Amount>
    <DisplayAmount></DisplayAmount>
    <FECAmount></FECAmount>
    <FECCurrencyCode></FECCurrencyCode>
    <Products/>
    <Addresses> <VPosAddressContract>
        <Type>1</Type>
        <Name>Mahmut Sami YAZAR</Name>
        <PhoneNumber>324234234234</PhoneNumber>
```

```
<OrderId>0</OrderId>
<AddressId>12</AddressId>
<Email>mahmutsamiyazar@hotmail.com</Email>
</VPosAddressContract>
</Addresses>
<APIVersion>1.0.0</APIVersion>
<PaymentType>None</PaymentType>
<DebtId>0</DebtId>
<SurchargeAmount>0</SurchargeAmount>
<SGKDebtAmount>0</SGKDebtAmount>
<InstallmentMaturityCommisionFlag>0</InstallmentMaturityCommisionFlag>
<TransactionSecurity>0</TransactionSecurity>
</VPosMessage>
<IsEnrolled>true</IsEnrolled>
<IsVirtual>false</IsVirtual>
<RRN>922811016643</RRN>
<Stan>016643</Stan>
<ResponseCode>05</ResponseCode>
<ResponseMessage>Hata Kodu5</ResponseMessage>
<OrderId>15184</OrderId>
<TransactionTime>2019-08-16T11:20:21.6283302</TransactionTime>
<MerchantOrderId>1771489024</MerchantOrderId>
<BusinessKey>0</BusinessKey>
</VPosTransactionResponseContract>
```

## Ön Provizyon Kapama İşlemi Örnek Kodlar

### Ön Provizyon Kapama İşleminin Yapılması

**Asp.Net**

[HttpPost]

```

public VPosTransactionResponseContract PreAuthorizationClose()
{
    VPosMessageContract request = new VPosMessageContract();
    request.MerchantId = MerchantId;
    request.CustomerId = CustomerId;
    request.UserName = UserName;
    request.MerchantOrderId = MerchantOrderId;
    request.OrderId = OrderId;
    //HashData Hesaplanması

    string hashPassword = "";
    string hashString = CreateHashString(merchantId: request.MerchantId, merchantOrderId:
    "", amount: default(int), okUrl: "", failUrl: "", username: request.UserName,
    hashPassword: hashPassword);
    request.HashData = ComputeHash(hashString);

    VPosTransactionResponseContract response = new VPosTransactionResponseContract();

    XmlSerializer x = new XmlSerializer(request.GetType());
    StringWriter sw = new StringWriter();
    x.Serialize(sw, request);
    XsdDataContractExporter exporter = new XsdDataContractExporter();
    string requestMessage = sw.ToString();

    string result =
    DataPost("https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/PreAuthorizationClose",
    requestMessage);
    try
    {
        response =
        DeserializeObject<VPosTransactionResponseContract>(result.Replace("\u0000", "").Replace(
        encoding="utf-16\"", ""));
    }
    catch (Exception ex)
    {
        string errorMessage = "Mesaj1: " + ex.Message + ": Mesaj2: " + ex.StackTrace + ":" +
        Mesaj3: " + ex.InnerException;
    }

    return response;
}

```

### 3.1.10. Ön Provizyon Kısmı Kapama İşlemi

#### Ön Provizyon Kısmı Kapama İşlemi Mesaj Yapısı

```

<?xml version="1.0" encoding="UTF-16"?>

<VPosMessageContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
<HashData> </HashData>
<MerchantId></MerchantId>
<SubMerchantId></SubMerchantId>
<CustomerId></CustomerId>
<UserName> </UserName>
<CustomerIPAddress></CustomerIPAddress>
<OrderId></OrderId>
<MerchantOrderId></MerchantOrderId>
<Amount></Amount>
<DisplayAmount></DisplayAmount>
</VPosMessageContract>
```

### Ön Provizyon Kısmı Kapama İşlemi Mesaj Örneği

```
<?xml version="1.0" encoding="UTF-16"?>
<VPosMessageContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<HashData>Wy0W65s/n4C3sImwLXzRfv+BwhU=</HashData>
<MerchantId>1</MerchantId>
<SubMerchantId>0</SubMerchantId>
<CustomerId>11111</CustomerId>
<UserName>APIUSER</UserName>
<HashPassword>kfkdsnskslkclswr9430ır</ HashPassword >
<OrderId>15184</OrderId>
<MerchantOrderId>1771489024</MerchantOrderId>
<Amount>100</Amount>
<DisplayAmount>100</DisplayAmount>
</VPosMessageContract>
```

## Ön Provizyon Kısmı Kapama İşlemi Sonucu Mesaj Yapısı

```
<?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>

    <HashData> </HashData>

    <MerchantId></MerchantId>

    <SubMerchantId></SubMerchantId>

    <CustomerId></CustomerId>

    <UserName> </UserName>

    <CustomerIPAddress></CustomerIPAddress>

    <OrderId></OrderId>

    <BatchId></BatchId>

    <MerchantOrderId></MerchantOrderId>

    <InstallmentCount></InstallmentCount>

    <Amount></Amount>

    <DisplayAmount></DisplayAmount>

    <FECAmount></FECAmount>

    <FECCurrencyCode></FECCurrencyCode>

    <Products/>

    <Addresses>

        <VPosAddressContract>

            <Type></Type>

            <Name> </Name>

            <PhoneNumber></PhoneNumber>

            <OrderId></OrderId>

            <AddressId></AddressId>

            <Email> </Email>

        </VPosAddressContract>

    </Addresses>

    <APIVersion></APIVersion>
```

```
<PaymentType> </PaymentType>
<SurchargeAmount></SurchargeAmount>
<SGKDebtAmount></SGKDebtAmount>
<InstallmentMaturityCommisionFlag></InstallmentMaturityCommisionFlag>
<TransactionSecurity></TransactionSecurity>
</VPosMessage>
<IsEnrolled> </IsEnrolled>
<IsVirtual> </IsVirtual>
<RRN></RRN>
<Stan></Stan>
<ResponseCode></ResponseCode>
<ResponseMessage> </ResponseMessage>
<OrderId></OrderId>
<TransactionTime></TransactionTime>
<MerchantOrderId></MerchantOrderId>
<BusinessKey></BusinessKey>
</VPosTransactionResponseContract>
```

## Ön Provizyon Kısmı Kapama İşlemi Örnek Kodlar

### Ön Provizyon Kısmı Kapama İşlemi Yapılması

#### Asp.Net

```
[HttpPost]
public VPosTransactionResponseContract PreAuthorizationPartialClose()
{
    VPosMessageContract request = new VPosMessageContract();
    request.MerchantId = "MerchantId";
    request.CustomerId = "CustomerId";
    request.UserName = "UserName";
    request.MerchantOrderId = "";
    request.OrderId = "";
    request.Amount = 100;
    request.DisplayAmount = "100";

    //HashData Hesaplanması

    string hashPassword = "";
```

```

        string hashString = CreateHashString(merchantId: request.MerchantId, merchantOrderId:
        "", amount: default(int), okUrl: "", failUrl: "", username: request.UserName,
        hashPassword: hashPassword);
        request.HashData = ComputeHash(hashString);

        VPosTransactionResponseContract response = new VPosTransactionResponseContract();

        XmlSerializer x = new XmlSerializer(request.GetType());
        StringWriter sw = new StringWriter();
        x.Serialize(sw, request);
        XsdDataContractExporter exporter = new XsdDataContractExporter();
        string requestMessage = sw.ToString();

        string result =
DataPost("https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/PreAuthorizationPartialC
lose", requestMessage);
        try
        {
            response =
DeserialzeObject<VPosTransactionResponseContract>(result.Replace("\u00a0", "").Replace(
encoding="utf-16\"", ""));
        }
        catch (Exception ex)
        {
            string errorMessage = "Mesaj1: " + ex.Message + ": Mesaj2: " + ex.StackTrace + ":" +
Mesaj3: " + ex.InnerException;
        }

        return response;
    }
}

```

### 3.1.11. Ön Provizyon İptal İşlemi

#### Ön Provizyon İptal İşlemi Mesaj Yapısı

```

<?xml version="1.0" encoding="UTF-16"?>

<VPosMessageContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <HashData> </HashData>

    <MerchantId></MerchantId>

    <SubMerchantId></SubMerchantId>

    <CustomerId></CustomerId>

    <UserName> </UserName>

    <CustomerIPAddress></CustomerIPAddress>

    <OrderId></OrderId>

```

```
<MerchantOrderId></MerchantOrderId>  
</VPosMessageContract>
```

## Ön Provizyon İptal İşlemi Mesaj Örneği

```
<?xml version="1.0" encoding="UTF-16"?>  
<VPosMessageContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
    <HashData>t6XJZeNvT+wKtj4GsHZvodG9y3E=</HashData>  
    <MerchantId>1</MerchantId>  
    <SubMerchantId>0</SubMerchantId>  
    <CustomerId>11111</CustomerId>  
    <UserName>APIUSER</UserName>  
    <HashPassword>kfkdsnskslkclswr9430ır</ HashPassword >  
    <OrderId>15184</OrderId>  
    <MerchantOrderId>1771489024</MerchantOrderId>  
</VPosMessageContract>
```

## Ön Provizyon İşlemi Sonucu Mesaj Yapısı

```
<?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>  
    <HashData></HashData>  
    <MerchantId></MerchantId>  
    <SubMerchantId></SubMerchantId>  
    <CustomerId></CustomerId>  
    <UserName> </UserName>  
    <CustomerIPAddress></CustomerIPAddress>  
    <OrderId></OrderId>  
    <BatchId></BatchId>  
    <MerchantOrderId></MerchantOrderId>
```

```
<InstallmentCount></InstallmentCount>
<Amount></Amount>
<DisplayAmount></DisplayAmount>
<FECAmount></FECAmount>
<FECCurrencyCode></FECCurrencyCode>
<Products/>
<Addresses>
    <VPosAddressContract>
        <Type></Type>
        <Name> </Name>
        <PhoneNumber></PhoneNumber>
        <OrderId></OrderId>
        <AddressId></AddressId>
        <Email> </Email>
    </VPosAddressContract>
</Addresses>
<APIVersion></APIVersion>
<PaymentType> </PaymentType>
<SurchargeAmount></SurchargeAmount>
<SGKDebtAmount></SGKDebtAmount>
<InstallmentMaturityCommisionFlag></InstallmentMaturityCommisionFlag>
<TransactionSecurity></TransactionSecurity>
</VPosMessage>
<IsEnrolled> </IsEnrolled>
<IsVirtual> </IsVirtual>
<RRN></RRN>
<Stan></Stan>
<ResponseCode></ResponseCode>
<ResponseMessage> </ResponseMessage>
<OrderId></OrderId>
```

```
<TransactionTime></TransactionTime>  
<MerchantOrderId></MerchantOrderId>  
<BusinessKey></BusinessKey>  
</VPosTransactionResponseContract>
```

## Ön Provizyon İptal İşlemi Örnek Kodlar

### Ön Provizyon İşlemi Yapılması

#### Asp.Net

```
[HttpPost]  
public VPosTransactionResponseContract PreAuthorizationReversal()  
{  
    VPosMessageContract request = new VPosMessageContract();  
    request.MerchantId = "MerchantId";  
    request.CustomerId = "CustomerId";  
    request.UserName = "UserName";  
    request.MerchantOrderId = "MerchantOrderId";  
    request.OrderId = "OrderId";  
    //HashData Hesaplanması  
  
    string hashPassword = "";  
    string hashString = CreateHashString(merchantId: request.MerchantId, merchantOrderId:  
    "", amount: default(int), okUrl: "", failUrl: "", username: request.UserName,  
    hashPassword: hashPassword);  
    request.HashData = ComputeHash(hashString);  
  
    VPosTransactionResponseContract response = new VPosTransactionResponseContract();  
  
    XmlSerializer x = new XmlSerializer(request.GetType());  
    StringWriter sw = new StringWriter();  
    x.Serialize(sw, request);  
    XsdDataContractExporter exporter = new XsdDataContractExporter();  
    string requestMessage = sw.ToString();  
  
    string result =  
    DataPost("https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/PreAuthorizationReversal  
    ", requestMessage);  
    try  
    {  
        response =  
        DeserializeObject<VPosTransactionResponseContract>(result.Replace("\x0", "").Replace("encoding=utf-16\"", ""));  
    }  
    catch (Exception ex)  
    {
```

```
        string errorMessage = "Mesaj1: " + ex.Message + ": Mesaj2: " + ex.StackTrace + ":"  
Mesaj3: " + ex.InnerException;  
    }  
  
    return response;  
}
```

### 3.1.12. Ön Provizyon İade İşlemi

#### Ön Provizyon İade İşlemi Mesaj Yapısı

```
<?xml version="1.0" encoding="UTF-16"?>  
  
<VPosMessageContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
  
    <HashData> </HashData>  
  
    <MerchantId></MerchantId>  
  
    <SubMerchantId></SubMerchantId>  
  
    <CustomerId></CustomerId>  
  
    <UserName> </UserName>  
  
    <CustomerIPAddress></CustomerIPAddress>  
  
    <OrderId></OrderId>  
  
    <MerchantOrderId></MerchantOrderId>  
  
</VPosMessageContract>
```

#### Ön Provizyon İade İşlemi Mesaj Örneği

```
<?xml version="1.0" encoding="UTF-16"?>  
  
<VPosMessageContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
  
    <HashData>t6XJZeNvT+wKtj4GsHZvodG9y3E=</HashData>  
  
    <MerchantId>1</MerchantId>  
  
    <SubMerchantId>0</SubMerchantId>  
  
    <CustomerId>11111</CustomerId>  
  
    <UserName>APIUSER</UserName>  
  
    <HashPassword>kfkdsnskslkclswr9430ır</ HashPassword >  
  
    <OrderId>15184</OrderId>  
  
    <MerchantOrderId>1771489024</MerchantOrderId>
```

</VPosMessageContract>

## Ön Provizyon İade İşlemi Sonucu Mesaj Yapısı

```
<?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>

    <HashData> </HashData>

    <MerchantId></MerchantId>

    <SubMerchantId></SubMerchantId>

    <CustomerId></CustomerId>

    <UserName> </UserName>

    <CustomerIPAddress></CustomerIPAddress>

    <OrderId></OrderId>

    <BatchId></BatchId>

    <MerchantOrderId></MerchantOrderId>

    <InstallmentCount></InstallmentCount>

    <Amount></Amount>

    <DisplayAmount></DisplayAmount>

    <FECAmount></FECAmount>

    <FECCurrencyCode></FECCurrencyCode>

    <Products/>

    <Addresses>

        <VPosAddressContract>

            <Type></Type>

            <Name> </Name>

            <PhoneNumber></PhoneNumber>

            <OrderId></OrderId>

            <AddressId></AddressId>

            <Email> </Email>

        </VPosAddressContract>

    </Addresses>

</VPosMessage>
```

```

    </Addresses>
    <APIVersion></APIVersion>
    <PaymentType> </PaymentType>
    <SurchargeAmount></SurchargeAmount>
    <SGKDebtAmount></SGKDebtAmount>
    <InstallmentMaturityCommisionFlag></InstallmentMaturityCommisionFlag>
    <TransactionSecurity></TransactionSecurity>

</VPosMessage>
<IsEnrolled> </IsEnrolled>
<IsVirtual> </IsVirtual>
<RRN></RRN>
<Stan></Stan>
<ResponseCode></ResponseCode>
<ResponseMessage> </ResponseMessage>
<OrderId>1514</OrderId>
<TransactionTime></TransactionTime>
<MerchantOrderId></MerchantOrderId>
<BusinessKey></BusinessKey>
</VPosTransactionResponseContract>

```

## Ön Provizyon İade İşlemi Örnek Kodlar

### Ön Provizyon İade İşlemi Yapılması

#### Asp.Net

```

[HttpPost]
public VPosTransactionResponseContract PreAuthorizationDrawBack()
{
    VPosMessageContract request = new VPosMessageContract();
    request.MerchantId = "MerchantId";
    request.CustomerId = "CustomerId";
    request.UserName = "UserName";
    request.MerchantOrderId = "";
    request.OrderId = "";
    //HashData Hesaplanması

    string hashPassword = "";
}

```

```

        string hashString = CreateHashString(merchantId: request.MerchantId, merchantOrderId:
        "", amount: default(int), okUrl: "", failUrl: "", username: request.UserName,
hashPassword: hashPassword);
request.HashData = ComputeHash(hashString);

VPosTransactionResponseContract response = new VPosTransactionResponseContract();

XmlSerializer x = new XmlSerializer(request.GetType());
StringWriter sw = new StringWriter();
x.Serialize(sw, request);
XsdDataContractExporter exporter = new XsdDataContractExporter();
string requestMessage = sw.ToString();

string result =
DataPost("https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/PreAuthorizationDrawBack
", requestMessage);
try
{
    response =
DeserialzeObject<VPosTransactionResponseContract>(result.Replace("\u00a0", "").Replace(
encoding=\\"utf-16\\\"", ""));
}
catch (Exception ex)
{
    string errorMessage = "Mesaj1: " + ex.Message + ": Mesaj2: " + ex.StackTrace + ":" +
Mesaj3: " + ex.InnerException;
}

return response;
}

```

## 3.2. Saklanan Kart İşlemleri Mesaj Yapıları

### 3.2.1. CardSave İşlemi

#### CardSave İşlemi Mesaj Yapısı

##### Sigorta firması olmayan üye işyerleri

```

<?xml version="1.0" encoding="utf-16"?>
<VPosCardDataContract xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<HashData></HashData>
<MerchantId></MerchantId>
<SubMerchantId></SubMerchantId>
<CustomerId></CustomerId>

```

```
<UserName></UserName>
<CardNumber></CardNumber>
<CardExpireDateYear></CardExpireDateYear>
<CardExpireDateMonth></CardExpireDateMonth>
<CardCVV2></CardCVV2>
<CardHolderName> </CardHolderName>
</VPosCardDataContract>
```

### Sigorta firması üye işyerleri

```
<?xml version="1.0" encoding="utf-16"?>
<VPosCardDataContract xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<HashData></HashData>
<MerchantId></MerchantId>
<SubMerchantId></SubMerchantId>
<CustomerId></CustomerId>
<UserName> </UserName>
<CardNumber></CardNumber>
<CardHolderName> </CardHolderName>
<IdentityTaxNumber></IdentityTaxNumber>
</VPosCardDataContract>
```

### CardSave İşlemi Örnek Mesaj

### Sigorta firması olmayan üye işyerleri

```
<?xml version="1.0" encoding="utf-16"?>
<VPosCardDataContract xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<HashData>3nBmkOJx12341234wgQFia8znl0=</HashData>
<MerchantId>1</MerchantId>
```

```
<SubMerchantId>0</SubMerchantId>  
<CustomerId>11111</CustomerId>  
<UserName>testUser</UserName>  
<CardNumber>5353550000951234</CardNumber>  
<CardExpireDateYear>23</CardExpireDateYear>  
<CardExpireDateMonth>01</CardExpireDateMonth>  
<CardCVV2>123</CardCVV2>  
<CardHolderName>Hasan Karacan</CardHolderName>  
</VPosCardDataContract>
```

### Sigorta firması üye işyerleri

```
<?xml version="1.0" encoding="utf-16"?>  
<VPosCardDataContract xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  
  <HashData>3nBmkOJx12341234wgQFia8znl0=</HashData>  
  <MerchantId>1</MerchantId>  
  <SubMerchantId>0</SubMerchantId>  
  <CustomerId>11111</CustomerId>  
  <UserName>TESTUSER</UserName>  
  <CardNumber>123456*1234</CardNumber>  
  <CardHolderName>Hasan Karacan</CardHolderName>  
  <IdentityTaxNumber>12345678902</IdentityTaxNumber>  
</VPosCardDataContract>
```

### CardSave İşlemi Sonucu Mesaj Yapısı

```
<?xml version="1.0" encoding="utf-16"?>  
<VPosTransactionResponseContract xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
<ResponseCode>00</ResponseCode>
<ResponseMessage />
<CardGuid>429675215AD44D38B05793911EAE1F9D</CardGuid>
</VPosTransactionResponseContract>
```

## CardSave İşlemi Örnek Kodlar

```
public VPosTransactionResponseContract CardSave(object model)
{
    VPosCardDataContract request = new VPosCardDataContract();
    request.MerchantId = 1;
    request.SubMerchantId = 0;
    request.CustomerId = 11111;
    request.UserName = "UserName";
    request.CardNumber = model.VPosCardData.CardNumber;
    request.CardExpireDateMonth = model.VPosCardData.CardExpireDateMonth;
    request.CardExpireDateYear = model.VPosCardData.CardExpireDateYear;
    request.CardCVV2 = model.VPosCardData.CardCVV2;
    request.CardHolderName = model.VPosCardData.CardHolderName;
    request.IdentityTaxNumber = model.VPosCardData.IdentityTaxNumber;

    //HashData Hesaplanması
    string hashPassword = "";
    string hashString = CreateHashString(merchantId: request.MerchantId, merchantOrderId: "", amount: default(int), okUrl: "", failUrl: "", username: request.UserName, hashPassword: hashPassword);
    request.HashData = ComputeHash(hashString);

    VPosTransactionResponseContract response = new VPosTransactionResponseContract();

    XmlSerializer x = new XmlSerializer(request.GetType());
    StringWriter sw = new StringWriter();
    x.Serialize(sw, request);
    XsdDataContractExporter exporter = new XsdDataContractExporter();
    string requestMessage = sw.ToString();

    string result =
    DataPost("https://boa.vakifkatlim.com.tr/VirtualPOS.Gateway/Home/CardSave",
    requestMessage);
    try
    {
        response =
        DeserializeObject<VPosTransactionResponseContract>(result.Replace("\u00a0", "").Replace("encoding=utf-16\"", ""));
    }
```

```
    catch (Exception ex)
    {
        string errorMessage = "Mesaj1: " + ex.Message + ": Mesaj2: " + ex.StackTrace + ":" + ex.InnerException;
    }

    return response;
}
```

### 3.2.2. CardSelect İşlemi

#### CardSelect İşlemi Mesaj Yapısı

```
<?xml version="1.0" encoding="utf-16"?>

<VPosCardDataContract xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

    <HashData></HashData>

    <MerchantId></MerchantId>

    <SubMerchantId></SubMerchantId>

    <CustomerId></CustomerId>

    <UserName></UserName>

    <CardGuid></CardGuid>

</VPosCardDataContract>
```

#### CardSelect İşlemi Örnek Mesaj

```
<?xml version="1.0" encoding="utf-16"?>

<VPosTransactionResponseContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <VPosCardData>

        <VposCardInfoContract>

            <VposCardInfoContract>

                <CardGuid>AAB4BBC307654E6CB4BEB3BB89ADDB61</CardGuid>

                <CardNumber>5406*****6038</CardNumber>

                <CardHolderName />
```

```
<SystemDate>2020-12-10T15:14:16.34</SystemDate>
</VposCardInfoContract>
</VposCardInfoContract>
</VPosCardData>
<ResponseCode>00</ResponseCode>
<ResponseMessage />
</VPosTransactionResponseContract>
```

### **CardSelect İşlemi Sonucu Mesaj Yapısı**

```
<?xml version="1.0" encoding="utf-16"?>
<VPosTransactionResponseContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <VPosCardData>
    <VposCardInfoContract>
      <VposCardInfoContract>
        <CardGuid>AAB4BBC307654E6CB4BEB3BB89ADDB61</CardGuid>
        <CardNumber>5406*****6038</CardNumber>
        <CardHolderName />
        <SystemDate>2020-12-10T15:14:16.34</SystemDate>
      </VposCardInfoContract>
      <VposCardInfoContract>
        <CardGuid>471594AB89BE4295B33DECFD741BBBCB</CardGuid>
        <CardNumber>5353*****8906</CardNumber>
        <CardHolderName>Hasan Karacan</CardHolderName>
        <SystemDate>2020-12-18T17:09:44.987</SystemDate>
      </VposCardInfoContract>
      <VposCardInfoContract>
        <CardGuid>825F9895CA984FF58922F8194BB622FF</CardGuid>
        <CardNumber>5353*****2302</CardNumber>
      </VposCardInfoContract>
    </VposCardInfoContract>
  </VPosCardData>
</VPosTransactionResponseContract>
```

```

<CardHolderName>Hasan Karacan</CardHolderName>
<SystemDate>2020-12-18T17:20:40.56</SystemDate>
</VposCardInfoContract>
</VposCardInfoContract>
</VposCardInfoContract>
</VPosCardData>
<ResponseCode>00</ResponseCode>
<ResponseMessage />
</VPosTransactionResponseContract>

```

### CardSelect İşlemi Örnek Kodlar

```

public VPosTransactionResponseContract CardSelect()
{
    VPosCardDataContract request = new VPosCardDataContract();
    request.MerchantId = 1;
    request.SubMerchantId = 0;
    request.CustomerId = 11111;
    request.UserName = "UserName";
    request.CardGuid = "gkdkmgasdjklsjklfskl";
    //HashData Hesaplanması
    string hashPassword = "";
    string hashString = CreateHashString(merchantId: request.MerchantId, merchantOrderId:
    "", amount: default(int), okUrl: "", failUrl: "", username: request.UserName,
    hashPassword: hashPassword);
    request.HashData = ComputeHash(hashString);

    VPosTransactionResponseContract response = new VPosTransactionResponseContract();

    XmlSerializer x = new XmlSerializer(request.GetType());
    StringWriter sw = new StringWriter();
    x.Serialize(sw, request);
    XsdDataContractExporter exporter = new XsdDataContractExporter();
    string requestMessage = sw.ToString();

    string result =
    DataPost("https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/CardSelect",
    requestMessage);
    try
    {
        response =
        DeserializeObject<VPosTransactionResponseContract>(result.Replace("\u00a0", "").Replace("encoding=utf-16", ""));
    }
}

```

```
        }
    catch (Exception ex)
    {
        string errorMessage = "Mesaj1: " + ex.Message + ": Mesaj2: " + ex.StackTrace + ":";  
        Mesaj3: " + ex.InnerException;
    }

    return response;
}
```

### 3.2.3. CardSelectAll İşlemi

#### CardSelectAll İşlemi Mesaj Yapısı

```
<?xml version="1.0" encoding="utf-16"?>

<VPosCardDataContract xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<HashData></HashData>

<MerchantId></MerchantId>

<SubMerchantId></SubMerchantId>

<CustomerId></CustomerId>

<UserName></UserName>

<BeginDate></BeginDate>

<EndDate></EndDate>

</VPosCardDataContract>
```

#### CardSelectAll İşlemi Örnek Mesaj

```
<?xml version="1.0" encoding="utf-16"?>

<VPosCardDataContract xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<HashData>111</HashData>

<MerchantId>1</MerchantId>

<SubMerchantId>0</SubMerchantId>

<CustomerId>11111</CustomerId>
```

```
<UserName>testUser</UserName>
<BeginDate>2020-09-08T00:34:00.000</BeginDate>
<EndDate>2020-09-08T00:34:00.000</EndDate>
</VPosCardDataContract>
```

### CardSelectAll İşlemi Sonucu Mesaj Yapısı

```
<?xml version="1.0" encoding="utf-16"?>
<VPosTransactionResponseContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <VPosCardData>
    <VposCardInfoContract>
      <VposCardInfoContract>
        <CardGuid>AAB4BBC307654E6CB4BEB3BB89ADDB61</CardGuid>
        <CardNumber>5406*****6038</CardNumber>
        <CardHolderName />
        <SystemDate>2020-12-10T15:14:16.34</SystemDate>
      </VposCardInfoContract>
      <VposCardInfoContract>
        <CardGuid>471594AB89BE4295B33DECFD741BBBCB</CardGuid>
        <CardNumber>5353*****8906</CardNumber>
        <CardHolderName>Hasan Karacan</CardHolderName>
        <SystemDate>2020-12-18T17:09:44.987</SystemDate>
      </VposCardInfoContract>
      <VposCardInfoContract>
        <CardGuid>825F9895CA984FF58922F8194BB622FF</CardGuid>
        <CardNumber>5353*****2302</CardNumber>
        <CardHolderName>Hasan Karacan</CardHolderName>
        <SystemDate>2020-12-18T17:20:40.56</SystemDate>
      </VposCardInfoContract>
    </VposCardInfoContract>
  </VPosCardData>
</VPosTransactionResponseContract>
```

```
</VposCardInfoContract>
</VposCardInfoContract>
</VposCardInfoContract>
</VPosCardData>
<ResponseCode>00</ResponseCode>
<ResponseMessage />
</VPosTransactionResponseContract>
```

## CardSelectAll İşlemi Örnek Kodlar

```
public VPosTransactionResponseContract CardSelectAll()
{
    VPosCardDataContract request = new VPosCardDataContract();
    request.MerchantId = 1;
    request.SubMerchantId = 0;
    request.CustomerId = 11111;
    request.UserName = "UserName";
    model.VPosCardData.BeginDate = DateTime.Now.AddMonths(-1).Date;
    model.VPosCardData.EndDate = DateTime.Now.Date;
    //HashData Hesaplanması

    string hashPassword = "";
    string hashString = CreateHashString(merchantId: request.MerchantId, merchantOrderId:
    "", amount: default(int), okUrl: "", failUrl: "", username: request.UserName,
    hashPassword: hashPassword);
    request.HashData = ComputeHash(hashString);

    VPosTransactionResponseContract response = new VPosTransactionResponseContract();

    XmlSerializer x = new XmlSerializer(request.GetType());
    StringWriter sw = new StringWriter();
    x.Serialize(sw, request);
    XsdDataContractExporter exporter = new XsdDataContractExporter();
    string requestMessage = sw.ToString();

    string result =
    DataPost("https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/CardSelectAll",
    requestMessage);
    try
    {
        response =
        DeserializeObject<VPosTransactionResponseContract>(result.Replace("\u00a0", "").Replace(
        encoding="utf-16\"", ""));
```

```
        }
    catch (Exception ex)
    {
        string errorMessage = "Mesaj1: " + ex.Message + ": Mesaj2: " + ex.StackTrace + ":";  
        Mesaj3: " + ex.InnerException;
    }

    return response;
}
```

### 3.2.4. CardDelete İşlemi

#### CardDelete İşlemi Mesaj Yapısı

```
<?xml version="1.0" encoding="utf-16"?>

<VPosCardDataContract xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<HashData></HashData>

<MerchantId></MerchantId>

<SubMerchantId></SubMerchantId>

<CustomerId></CustomerId>

<UserName></UserName>

<CardGuid></CardGuid>

</VPosCardDataContract>
```

#### CardDelete İşlemi Örnek Mesaj

```
<?xml version="1.0" encoding="utf-16"?>

<VPosCardDataContract xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<HashData>111=</HashData>

<MerchantId>1</MerchantId>

<SubMerchantId>0</SubMerchantId>

<CustomerId>11111</CustomerId>

<UserName>testUser</UserName>
```

```
<CardGuid>A3193146D00E443985B2666A6C9754E2</CardGuid>  
</VPosCardDataContract>
```

### CardDelete İşlemi Sonucu Mesaj Yapısı

```
<?xml version="1.0" encoding="utf-16"?>  
<VPosTransactionResponseContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
    <ResponseCode>00</ResponseCode>  
    <ResponseMessage />  
</VPosTransactionResponseContract>
```

### CardDelete İşlemi Örnek Kodlar

```
public VPosTransactionResponseContract CardDelete()  
{  
    VPosCardDataContract request = new VPosCardDataContract();  
    request.MerchantId = 1;  
    request.SubMerchantId = 0;  
    request.CustomerId = 11111;  
    request.UserName = "UserName";  
    request.CardGuid = "gkdkmgasdjklsjklsjklfskl";  
    //HashData Hesaplanması  
    string hashPassword = "";  
    string hashString = CreateHashString(merchantId: request.MerchantId, merchantOrderId:  
    "", amount: default(int), okUrl: "", failUrl: "", username: request.UserName,  
    hashPassword: hashPassword);  
    request.HashData = ComputeHash(hashString);  
  
    VPosTransactionResponseContract response = new VPosTransactionResponseContract();  
  
    XmlSerializer x = new XmlSerializer(request.GetType());  
    StringWriter sw = new StringWriter();  
    x.Serialize(sw, request);  
    XsdDataContractExporter exporter = new XsdDataContractExporter();  
    string requestMessage = sw.ToString();  
  
    string result =  
    DataPost("https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home/CardDelete",  
    requestMessage);  
    try  
    {  
        response =  
        DeserializeObject<VPosTransactionResponseContract>(result.Replace("\#x0;", "").Replace(  
        "encoding=\\"utf-16\\\"", ""));  
    }
```

```
        catch (Exception ex)
    {
        string errorMessage = "Mesaj1: " + ex.Message + ": Mesaj2: " + ex.StackTrace + ":" + ex.InnerException;
    }

    return response;
}
```

### 3.3. Sipariş İşlemleri Mesaj Yapıları

#### 3.3.1. SelectOrder İşlemi

##### SelectOrder İşlemi Mesaj Yapısı

```
<?xml version="1.0" encoding="utf-16"?>

<VPosMessageContract xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<HashData></HashData>

<MerchantId></MerchantId>

<SubMerchantId></SubMerchantId>

<CustomerId></CustomerId>

<UserName> </UserName>

<StartDate></StartDate>

<EndDate></EndDate>

<LowerLimit></LowerLimit>

<UpperLimit></UpperLimit>

<ProvNumber></ProvNumber>

<OrderStatus></OrderStatus >

<TranResult></TranResult >

<OrderNo></OrderNo >

</VPosMessageContract>
```

##### SelectOrder İşlemi Örnek Mesaj

```
<?xml version="1.0" encoding="utf-16"?>
```

```
<VPosMessageContract xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<HashData>xxxxx=</HashData>

<MerchantId>1</MerchantId>

<SubMerchantId>0</SubMerchantId>

<CustomerId>11111</CustomerId>

<UserName>user</UserName>

<StartDate>2020-12-01</StartDate>

<EndDate>2020-12-25</EndDate>

<LowerLimit>0</LowerLimit>

<UpperLimit>100</UpperLimit>

<ProvNumber></ProvNumber>

<OrderStatus>1</OrderStatus >

<TranResult>1</TranResult >

<OrderNo></OrderNo >

</VPosMessageContract>
```

### SelectOrder İşlemi Sonucu Mesaj Yapısı

```
<?xml version="1.0" encoding="utf-16"?>

<VPosTransactionResponseContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<VPosOrderData>

<OrderContract>

<OrderId>12754</OrderId>

<MerchantOrderId>709834990</MerchantOrderId>

<MerchantId>1</MerchantId>

<PosTerminalId>111111</PosTerminalId>

<OrderStatus>1</OrderStatus>

<OrderStatusDescription>Satis</OrderStatusDescription>
```

```
<OrderType>1</OrderType>
<OrderTypeDescription>Pesin</OrderTypeDescription>
<TransactionStatus>2</TransactionStatus>
<TransactionStatusDescription>Basarisiz</TransactionStatusDescription>
<LastOrderStatus>1</LastOrderStatus>
<LastOrderStatusDescription>Satis</LastOrderStatusDescription>
<EndOfDayStatus>1</EndOfDayStatus>
<EndOfDayStatusDescription>Acik</EndOfDayStatusDescription>
<FEC>0949</FEC>
<FecDescription>TRY</FecDescription>
<TransactionSecurity>5</TransactionSecurity>
<TransactionSecurityDescription />
<CardHolderName>Hasan Karacan</CardHolderName>
<CardType>MasterCard</CardType>
<CardNumber>5353*****3233</CardNumber>
<OrderDate>2020-12-25T12:13:35.74</OrderDate>
<TranAmount>3.90</TranAmount >
<FirstAmount>3.90</FirstAmount>
<FECAmount>0.00</FECAmount>
<CancelAmount>0.00</CancelAmount>
<DrawbackAmount>0.00</DrawbackAmount>
<ClosedAmount>0.00</ClosedAmount>
<InstallmentCount>0</InstallmentCount>
<ResponseCode>05</ResponseCode>
<ResponseExplain>Hata Kodu5</ResponseExplain>
<ProvNumber />
<RRN>03611114146</RRN>
<Stan>012246</Stan>
<MerchantUserName>USERNAME</MerchantUserName>
```

```
<BatchId>73</BatchId>
</OrderContract>
<OrderContract>
  <OrderId>12753</OrderId>
  <MerchantOrderId>424636131</MerchantOrderId>
  <MerchantId>1</MerchantId>
  <PosTerminalId>111111</PosTerminalId>
  <OrderStatus>1</OrderStatus>
  <OrderStatusDescription>Satis</OrderStatusDescription>
  <OrderType>1</OrderType>
  <OrderTypeDescription>Pesin</OrderTypeDescription>
  <TransactionStatus>1</TransactionStatus>
  <TransactionStatusDescription>Basarili</TransactionStatusDescription>
  <LastOrderStatus>1</LastOrderStatus>
  <LastOrderStatusDescription>Satis</LastOrderStatusDescription>
  <EndOfDayStatus>2</EndOfDayStatus>
  <EndOfDayStatusDescription>Kapali</EndOfDayStatusDescription>
  <FEC>0949</FEC>
  <FecDescription>TRY</FecDescription>
  <TransactionSecurity>5</TransactionSecurity>
  <TransactionSecurityDescription />
  <CardHolderName>Hasan Karacan</CardHolderName>
  <CardType>MasterCard</CardType>
  <CardNumber>5353*****8906</CardNumber>
  <OrderDate>2020-12-25T08:41:40.947</OrderDate>
  <FirstAmount>2.70</FirstAmount>
  <FECAmount>0.00</FECAmount>
  <CancelAmount>0.00</CancelAmount>
  <DrawbackAmount>0.00</DrawbackAmount>
```

```

<ClosedAmount>0.00</ClosedAmount>
<InstallmentCount>0</InstallmentCount>
<ResponseCode>00</ResponseCode>
<ResponseExplain>Provizyon alındı.</ResponseExplain>
<ProvNumber>831168</ProvNumber>
<RRN>036008014143</RRN>
<Stan>014143</Stan>
<MerchantUserName>USERNAME</MerchantUserName>
<BatchId>72</BatchId>
</OrderContract>
</VPosOrderData>
<ResponseCode>00</ResponseCode>
<ResponseMessage />
</VPosTransactionResponseContract>

```

### SelectOrder İşlemi Örnek Kodlar

```

public VPosTransactionResponseContract SelectOrder()
{
    VPosMessageContract request = new VPosMessageContract();
    request.MerchantId = 1;
    request.SubMerchantId = 0;
    request.CustomerId = 11111;
    request.UserName = "UserName";
    request.StartDate = "2020-12-20";
    request.EndDate = "2020-12-25";
    request.UpperLimit = 0;
    request.LowerLimit = 100;
    request.ProvNumber = "";
    request.OrderStatus = "";
    request.TranResult = "";
    request.OrderNo = "";

    //HashData Hesaplanması
    string hashPassword = "";
    string hashString = CreateHashString(merchantId: request.MerchantId, merchantOrderId:
    "", amount: default(int), okUrl: "", failUrl: "", username: request.UserName,
    hashPassword: hashPassword);
}

```

```
request.HashData = ComputeHash(hashString);

VPosTransactionResponseContract response = new VPosTransactionResponseContract();

XmlSerializer x = new XmlSerializer(request.GetType());
StringWriter sw = new StringWriter();
x.Serialize(sw, request);
XsdDataContractExporter exporter = new XsdDataContractExporter();
string requestMessage = sw.ToString();

string result =
DataPost("https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home>SelectOrder",
requestMessage);
try
{
    response =
DeserializeObject<VPosTransactionResponseContract>(result.Replace("\u0000", "").Replace(
encoding="utf-16\"", ""));
}
catch (Exception ex)
{
    string errorMessage = "Mesaj1: " + ex.Message + ": Mesaj2: " + ex.StackTrace + ":" +
Mesaj3: " + ex.InnerException;
}

return response;
}
```

### 3.3.2. SelectOrderByMerchantOrderId İşlemi

#### SelectOrderByMerchantOrderId İşlemi Mesaj Yapısı

```
<?xml version="1.0" encoding="utf-16"?>

<VPosMessageContract xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<HashData></HashData>

<MerchantId></MerchantId>

<SubMerchantId></SubMerchantId>

<CustomerId></CustomerId>

<UserName> </UserName>

<MerchantOrderId></MerchantOrderId>

</VPosMessageContract>
```

## SelectOrderByMerchantOrderId İşlemi Örnek Mesaj

```
<?xml version="1.0" encoding="utf-16"?>

<VPosMessageContract xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<HashData>xxxx=</HashData>

<MerchantId>1</MerchantId>

<SubMerchantId>0</SubMerchantId>

<CustomerId>11111</CustomerId>

<UserName>user</UserName>

<MerchantOrderId>1995434716</MerchantOrderId>

</VPosMessageContract>
```

## SelectOrderByMerchantOrderId İşlemi Sonucu Mesaj Yapısı

```
<?xml version="1.0" encoding="utf-16"?>

<VPosTransactionResponseContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<VPosOrderData>

<OrderContract>

<OrderId>12743</OrderId>

<MerchantOrderId>1995434716</MerchantOrderId>

<MerchantId>1</MerchantId>

<PosTerminalId>111111</PosTerminalId>

<OrderStatus>1</OrderStatus>

<OrderStatusDescription />

<OrderType>1</OrderType>

<OrderTypeDescription />

<TransactionStatus>1</TransactionStatus>

<TransactionStatusDescription>Basarili</TransactionStatusDescription>

<LastOrderStatus>1</LastOrderStatus>

<LastOrderStatusDescription />
```

```
<EndOfDayStatus>1</EndOfDayStatus>
<EndOfDayStatusDescription>Acik</EndOfDayStatusDescription>
<FEC>0949</FEC>
<FecDescription>TRY</FecDescription>
<TransactionSecurity>1</TransactionSecurity>
<TransactionSecurityDescription>3d'siz işlem</TransactionSecurityDescription>
<CardHolderName>Hasan Karacan</CardHolderName>
<CardType>MasterCard</CardType>
<CardNumber>5353*****7017</CardNumber>
<OrderDate>2020-12-24T09:21:41.55</OrderDate>
<FirstAmount>9.30</FirstAmount>
<FECAmount>0.00</FECAmount>
<CancelAmount>0.00</CancelAmount>
<DrawbackAmount>0.00</DrawbackAmount>
<ClosedAmount>0.00</ClosedAmount>
<InstallmentCount>0</InstallmentCount>
<ResponseCode>00</ResponseCode>
<ResponseExplain>Provizyon alındı.</ResponseExplain>
<ProvNumber>043290</ProvNumber>
<RRN>035909014127</RRN>
<Stan>014127</Stan>
<MerchantUserName>USERNAME</MerchantUserName>
<BatchId>69</BatchId>
</OrderContract>
</VPosOrderData>
<ResponseCode>00</ResponseCode>
<ResponseMessage />
</VPosTransactionResponseContract>
```

## SelectOrderByMerchantOrderId İşlemi Örnek Kodlar

```
public VPosTransactionResponseContract SelectOrderByMerchantOrderId ()
{
    VPosMessageContract request = new VPosMessageContract();
    request.MerchantId = 1;
    request.SubMerchantId = 0;
    request.CustomerId = 11111;
    request.UserName = "UserName";
    request.MerchantOrderId = "12345";

    //HashData Hesaplanması

    string hashPassword = "";
    string hashString = CreateHashString(merchantId: request.MerchantId, merchantOrderId:
    "", amount: default(int), okUrl: "", failUrl: "", username: request.UserName,
    hashPassword: hashPassword);
    request.HashData = ComputeHash(hashString);

    VPosTransactionResponseContract response = new VPosTransactionResponseContract();

    XmlSerializer x = new XmlSerializer(request.GetType());
    StringWriter sw = new StringWriter();
    x.Serialize(sw, request);
    XsdDataContractExporter exporter = new XsdDataContractExporter();
    string requestMessage = sw.ToString();

    string result =
    DataPost("https://boa.vakifkatilim.com.tr/VirtualPOS.Gateway/Home>SelectOrderByMerchantOrd
    erId", requestMessage);
    try
    {
        response =
        DeserializeObject<VPosTransactionResponseContract>(result.Replace("\u0130", "").Replace(
        encoding="utf-16\"", ""));
    }
    catch (Exception ex)
    {
        string errorMessage = "Mesaj1: " + ex.Message + ": Mesaj2: " + ex.StackTrace + ":" +
        Mesaj3: " + ex.InnerException;
    }

    return response;
}
```

## 4. GEREKLİ BİLGİLER

### 4.1. Değişken Tipleri

Değişken tipleri aşağıda gösterilmiştir:

Alan İsimleri	Tip	Zorunlu Alan mı?	Açıklama
OkUrl	string	evet	İşlem sonuçlarının geleceği, işyeri tarafında başarılı cevap mesajını karşılayacak programın olduğu site adresi bu alana yazılmalıdır.
FailUrl	string	evet	İşlem sonuçlarının geleceği, işyeri tarafında başarısız cevap mesajını karşılayacak programın olduğu site adresi bu alana yazılmalıdır.
HashData	string	evet	Gelen işyeri ve kullanıcı datasının doğrulanması için kullanılan alandır. İşyeri tarafından dökümda belirtilen şekilde bir hash hesaplaması yapılarak üretilen değer bu alanda yollanılır. Banka tarafından gelen datalara göre hash değeri hesaplanarak bu alanda yollanılan değer ile karşılaştırılır. Değerlerin aynı olması durumunda işlemin devamına izin verilir.
OrderId	string		Api servisleri tarafından başarılı işlem sonucu dönen banka taraflı satış numarası
MerchantOrderId	string	evet	Üye işyeri tarafından belirlenen tekil satış numarası (50 karakter sınırlamasında)
MerchantId	int	evet	Üye işyeri tekil numarası
SubMerchantId	int	hayır	Alt üye işyeri numarası (yoksa 0 gönderilmeli)
CustomerId	int	evet	Müşteri Numarası
UserName	string	evet	Yönetim panelinden oluşturulan api rollü kullanıcı adı
HashPassword	string	evet	Api Kullanıcı Şifresinin Dökümda ki Hashleme metoduyla hashlenmiş halidir
CardNumber	string	evet	Müşteriden alınan kart bilgisi bu alana girilmelidir. Sigorta şirketi ödemelerinde kart numarasının ilk 6 ve son 4 hanesi girilmelidir (123456*7890).
CardExpireDateYear	string	evet	Kart son kullanma yıl bilgisinin yollandığı alandır.
CardExpireDateMonth	string	evet	Kart son kullanma ay bilgisinin yollandığı alandır.
CardCVV2	string	evet	Kartın CVV2 CVC2 3CSC bilgisinin yollandığı alandır.
CardHolderName	string	evet	Kart sahibi ad soyad
IdentityTaxNumber	string	hayır	Sigorta şirketleri sanal pos üzerinden işlem yaptıklarında bu alanda TCKN bilgisi veya VKN bilgisi dönmek zorundadır (TCKN: 12345678901, VKN: 1234567890).

CardGuid	String	Hayır	Kart saklama hizmeti kullanan üye işyerleri için kart bilgileri yerine bu değeri göndererek işlem yapılmasına olanak sağlar. Yoksa boş gönderilmeli.
InstallmentCount	byte	evet	Taksit Sayısı (0 veya 1: Tek Çekim)
Amount	int	evet	Satış Tutarı. Tutar bilgisi 100 ile çarpılarak gönderilecektir. (100: 1,00 TL / 150: 1,50 TL / 5555: 55,55 TL / 100005: 1.000,05 TL)
DisplayAmount	string	hayır	Herhangi bir müdahale edilmemiş orjinal satış tutarı
CurrencyCode	string	evet	İşlem yapılan kur kodunun gönderildiği alandır. 0949: TRY
FECCurrencyCode	int	evet	İşlem yapılan kur kodunun gönderildiği alandır. 0949: TRY
TransactionSecurity	string	evet	1: Moto, 3: 3DSecure, 5: Non3DSecure
PaymentType	int	hayır	Ödeme Tipi (1:Satış / 2:Fatura Satış / 3:Fatura Satış İptali) eğer hiç birşey gönderilmez ise default olarak satış işlemi kabul edilir. Fatura işlemleri için bu alan zorunludur.
StartDate	Datetime	evet	Sipariş servisi isteği kullanılır. Sipariş tarihi kriteri içindir.
EndDate	Datetime	evet	Sipariş servisi isteği kullanılır. Sipariş tarihi kriteri içindir.
UpperLimit	decimal	hayır	Sipariş servisi isteği kullanılır. Sipariş tutarı kriteri içindir.
LowerLimit	decimal	hayır	Sipariş servisi isteği kullanılır. Sipariş tutarı kriteri içindir.
ProvNumber	string	hayır	Sipariş servisi isteği kullanılır. Sipariş provizyon numarasıdır.
PosTerminalId	string		Sipariş servisi dönüş değerlerindendir. Sanal pos terminal numarasıdır.
OrderStatus	int		Sipariş servisi dönüş değerlerindendir. Siparişin ilk statüsüdür.(Satış veya ön provizyon) Description geri dönüş değerinde açıklaması yer almaktadır.
OrderType	int		Sipariş servisi dönüş değerlerindendir. Taksitli mi tek çekimmi bilgisini tutar. Description geri dönüş değerinde açıklaması yer almaktadır.
TransactionStatus	int		Sipariş servisi dönüş değerlerindendir. İşlemin başarılı durumunu döner. Description geri dönüş değerinde açıklaması yer almaktadır.
LastOrderStatus	int		Sipariş servisi dönüş değerlerindendir. İşlemin devam eden son statüsünü tutar. (iade iptal vs)

			Description geri dönüş değerinde açıklaması yer almaktadır.
EndOfDayStatus	int		Sipariş servisi dönüş değerlerindendir. Gün sonu alınıp alınmadığı bilgisini döner.
FirstAmount	decimal		Sipariş servisi dönüş değerlerindendir. İşlemin ilk gerçekleştiği tutar bilgisidir.
CancelAmount	decimal		Sipariş servisi dönüş değerlerindendir. Eğer satış işlemi iptal edilmişse burada ne kadarlık tutarın iptal edildiği bilgisi tutulur.
DrawbackAmount	decimal		Sipariş servisi dönüş değerlerindendir. Eğer satış işlemi iade edilmişse burada ne kadarlık tutarın iade edildiği bilgisi tutulur.
ClosedAmount	decimal		Sipariş servisi dönüş değerlerindendir. Eğer ön provizyon işlemiyse bu alanda kapanan tutarı gösterir.
RRN	string		Sipariş servisi dönüş değerlerindendir. İşlemin kayıt numarasıdır.
BatchId	decimal		Sipariş servisi dönüş değerlerindendir. Gün sonu kapama numarasıdır.
CardType	string		Kart şemasını döner. (Mastercard, Visa, Troy)
CustomerIPAddress	string	hayır	Api servislerine bağlantı kurulan ip adresi
Addresses	List	hayır	Siparişler için adres bilgilerini tutmak içindir. VPosAddressContract isimde model sınıfı tanımlanmalı
Type	string	hayır	VPosAddressContract sınıfı değişkeni
Name	string	hayır	VPosAddressContract sınıfı değişkeni
LastName	string	hayır	VPosAddressContract sınıfı değişkeni
GSMNumber	string	hayır	VPosAddressContract sınıfı değişkeni
PhoneNumber	string	hayır	VPosAddressContract sınıfı değişkeni
OrderId	int	hayır	VPosAddressContract sınıfı değişkeni
AddressId	int	hayır	VPosAddressContract sınıfı değişkeni
Text	string	hayır	VPosAddressContract sınıfı değişkeni
Email	string	hayır	VPosAddressContract sınıfı değişkeni
AdditionalDataList	List	hayır	Siparişler için ekstra bilgileri tutar VPosAdditionalData isimde model sınıfı tanımlanmalı
Key	string	hayır	VPosAdditionalData sınıfı değişkeni
Data	string	hayır	VPosAdditionalData sınıfı değişkeni
Description	string	hayır	VPosAdditionalData sınıfı değişkeni
SurchargeAmount	decimal	hayır	Sgk borç ödemeleri içindir. Aktif Değildir.
SGKDebtAmount	decimal	hayır	Sgk borç ödemeleri içindir. Aktif Değildir.

InstallmentMaturityCommisionFlag	byte	hayır	Taksit Sayısı Vade Farkı Komisyonu alınması içindir. Aktif Değildir.
----------------------------------	------	-------	---

## 4.2. Model Sınıfları

```

/// Bu sınıf siparişler için ekstra bilgilerin listesini tutar.
[Serializable]
public class VPosMessageContract
{
    public string OkUrl { get; set; }
    public string FailUrl { get; set; }
    public string HashData { get; set; }
    public int MerchantId { get; set; }
    public int SubMerchantId { get; set; }
    public int CustomerId { get; set; }
    public string UserName { get; set; }
    public string MerchantOrderId { get; set; }
    public int InstallmentCount { get; set; }
    public int Amount { get; set; }
    public int FECAMount { get; set; }
    public string DisplayAmount { get; set; }
    public string FECCurrencyCode { get; set; }
    public string CurrencyCode { get; set; }
    public VPosAdditionalDataSet AdditionalData { get; set; }
    public List<VPosAddressContract> Addresses { get; set; }
    public string CardNumber { get; set; }
    public string CardExpireDateYear { get; set; }
    public string CardExpireDateMonth { get; set; }
    public string CardCVV2 { get; set; }
    public string CardHolderName { get; set; }
    public int PaymentType { get; set; }

    /// Order Servisleri Alanları
    public DateTime StartDate { get; set; }
    public DateTime EndDate { get; set; }
    public decimal? UpperLimit { get; set; }
    public decimal? LowerLimit { get; set; }
    public string ProvNumber { get; set; }
}

/// Bu sınıf siparişler için ekstra bilgilerin listesini tutar.
[Serializable]
public class VPosAdditionalDataSet
{
    public List<VPosAdditionalData> AdditionalDataList { get; set; }
}

/// Bu sınıf siparişler için ekstra bilgileri tutar.

```

```
[Serializable]
public class VPosAdditionalData
{
    public string Key { get; set; }
    public string Data { get; set; }
    public string Description { get; set; }
}

/// Bu sınıf siparişler için adres bilgilerini tutar.
[Serializable]
public class VPosAddressContract
{
    public string Type { get; set; }
    public string Name { get; set; }
    public string LastName { get; set; }
    public string Company { get; set; }
    public string Text { get; set; }
    public string District { get; set; }
    public string City { get; set; }
    public string Country { get; set; }
    public string PostalCode { get; set; }
    public string PhoneNumber { get; set; }
    public string GSMNumber { get; set; }
    public string FaxNumber { get; set; }
    public string Email { get; set; }
}

/// Bu sınıf sanal pos işlem bilgilerinin ve sonuçlarının tutulduğu değişkenleri içeren
/// sınıfıdır.
[Serializable]
public class VPosTransactionResponseContract
{
    public VPosMessageContract VPosMessage { get; set; }
    public VPosCardDataContract VPosCardData { get; set; }
    public List<OrderContract> VPosOrderData { get; set; }

    // eğer 3d secure ise geriye bu response dönülür.
    public bool IsEnrolled { get; set; }
    public bool IsVirtual { get; set; }
    public string PareqHtmlFormString { get; set; }

    // eğer işlem 3dsecure değilse.
    public string ProvisionNumber { get; set; }
    public string RRN { get; set; }
    public string Stan { get; set; }
    public string ResponseCode { get; set; }
    public bool IsSuccess { get; set; }
    public string ResponseMessage { get; set; }
    public int OrderId { get; set; }
    public DateTime TransactionTime { get; set; }
    public string MerchantOrderId { get; set; }
}
```

```
public string HashData { get; set; }
public string MD { get; set; }
public string ReferenceId { get; set; }
}

//Kart Servisleri için
[Serializable]
public class VPosCardDataContract : VPosBaseContract
{
    public List<VposCardInfoContract> VposCardInfoContract { get; set; }
    public DateTime? BeginDate { get; set; }
    public DateTime? EndDate { get; set; }
    public string CardGuid { get; set; }
}

[Serializable]
public partial class VposCardInfoContract
{
    public string CardGuid { get; set; }
    public string CardNumber { get; set; }
    public string CardHolderName { get; set; }
    public DateTime? SystemDate { get; set; }
}

[Serializable]
public class VPosBaseContract
{
    public string HashData { get; set; }
    public int MerchantId { get; set; }
    public int SubMerchantId { get; set; }
    public int CustomerId { get; set; }
    public string UserName { get; set; }
    public string CardNumber { get; set; }
    public string CardExpireDateYear { get; set; }
    public string CardExpireDateMonth { get; set; }
    public string CardCVV2 { get; set; }
    public string CardHolderName { get; set; }
    [StringLength(11)]
    public string IdentityTaxNumber { get; set; }
}

//Sipariş Servisleri için
[Serializable]
public partial class OrderContract
{
    public int OrderId { get; set; }
    public string MerchantOrderId { get; set; }
    public int MerchantId { get; set; }
    public string PosTerminalId { get; set; }
```

```
public int OrderStatus { get; set; }
public string OrderStatusDescription { get; set; }
public int OrderType { get; set; }
public string OrderTypeDescription { get; set; }
public int TransactionStatus { get; set; }
public string TransactionStatusDescription { get; set; }
public int? LastOrderStatus { get; set; }
public string LastOrderStatusDescription { get; set; }
public int? EndOfDayStatus { get; set; }
public string EndOfDayStatusDescription { get; set; }
public string FEC { get; set; }
public string FecDescription { get; set; }
public int? TransactionSecurity { get; set; }
public string TransactionSecurityDescription { get; set; }
public string CardHolderName { get; set; }
public string CardType { get; set; }
public string CardNumber { get; set; }
public DateTime OrderDate { get; set; }
public decimal FirstAmount { get; set; }
public decimal FECAmount { get; set; }
public decimal? CancelAmount { get; set; }
public decimal? DrawbackAmount { get; set; }
public decimal? ClosedAmount { get; set; }
public int InstallmentCount { get; set; }
public string ResponseCode { get; set; }
public string ResponseExplain { get; set; }
public string ProvNumber { get; set; }
public string RRN { get; set; }
public string Stan { get; set; }
public string MerchantUserName { get; set; }
public Int64 BatchId { get; set; }
}
```

### 4.3. Örnek Kodlarda ki Yardımcı Methodlar

```
public static T DeserializeObject<T>(string serializedObject) where T : class
{
    XmlSerializer xmlSerializer = new XmlSerializer(typeof(T));
    MemoryStream memoryStream = new MemoryStream(new
    UTF8Encoding().GetBytes(serializedObject));
    XmlTextWriter xmlTextWriter = new XmlTextWriter((Stream)memoryStream, Encoding.UTF8);
    return (T)xmlSerializer.Deserialize((Stream)memoryStream);
}
```

## 4.4. Hash Data Hesaplaması

Gelen cevabın Vakıf Katılım Sanal POS'u tarafından gönderildiğinin kontrol edilebilmesi için HashData değeri yer almaktadır. HashData alanının, üye işyerinin kendi oluşturduğu HashData ile aynı değere sahip olması bu cevap mesajının Vakıf Katılım Sanal POS'u tarafından gönderildiğini ifade etmektedir.

Hash hesaplamasında kullanılan parametreler şöyledir:

- MerchantId
- MerchantOrderID
- Amount
- OkUrl
- FailUrl
- UserName
- hashPassword

### 4.4.1. Hash data hesaplamalarında kullanılacak methodlar:

- Hash string oluşturmak için

```
public static String CreateHashString(int merchantId, int amount, string merchantOrderId = "", string okUrl = "", string failUrl = "", string username = "", string hashPassword = "")  
{  
    string newHash = string.Concat(merchantId  
        , merchantOrderId  
        , amount == default(int) ? "" : amount.ToString()  
        , okUrl  
        , failUrl  
        , username  
        , hashPassword);  
  
    return newHash;  
}
```

- **hashPassword** değerini hesaplamak için Api kullanıcısı şifresini ComputeHash methodundan geçirilmek gerekmektedir.

```
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.ToString(inputbytes, 16);
return hash;
}
```

**NOT:** Ayrıca dikkat edilmesi gereken bir başka nokta ise hash data hesaplamasında kullanılan alanlar her api servisinde gönderilmesi gerekmıyor ise (örn: OkUrl ve FailUrl kart servisleri için geçerli değildir.) hash string oluştururken bu alanları eklememek gerekmektedir.

#### 4.4.2. Hash Data Hesaplamasında Hata Alınması Halinde

Hash Data Hesaplamasında alınan hatalar hatalı veri girilmesinden kaynaklanmaktadır. Herhangi bir hata alınması durumunda aşağıdaki adımların ve bilgilerin kontrol edilmesi gerekmektedir.

**MerchantId** : Mağaza numarası girilmelidir.

**MerchantOrderId** : Sipariş numarası girilmelidir.

**Amount** : Tutarda nokta veya virgül bulundurulmamalıdır. Tutar bilgisi 100 ile çarpılmalıdır.

**OkUrl** : Başarılı sayfa yazılmalıdır ve başarısız sayfa ile aynı adres olmamalıdır.

**FailUrl** : Başarısız sayfa yazılmalıdır ve başarılı sayfa ile aynı adres olmamalıdır.

**UserName** : <https://kurumsal.vakifkatilim.com.tr> adresi üzerinden oluşturduğunuz api kullanıcı adı yazılmalıdır.

**HashPassword** : <https://kurumsal.vakifkatilim.com.tr> adresi üzerinden oluşturduğunuz api kullanıcı şifresinin hashlenmiş hali olmalıdır.

### 4.5. Api Servisleri İstek Gönderilmesi ve Geri Dönüş Cevabının Alınması

#### 4.5.1. İstek Post Edilmesi İçin Method

Yukarıda ki örnek kodlarda da yer alan isteğin gönderildiği DataPost methodu

```
using System.Net;
using System.IO;
private static string DataPost(string uriToPost, string dataToPost)
```

```
{
    byte[] buffer = Encoding.UTF8.GetBytes(dataToPost);
    ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12 |
    SecurityProtocolType.Ssl3 | SecurityProtocolType.Tls |
    SecurityProtocolType.Tls11;
    HttpWebRequest WebReq = (HttpWebRequest)WebRequest.Create(uriToPost);
    WebReq.Timeout = 50 * 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;
}
```

#### 4.5.2. Cevabın Alınması

Sanal pos api servislerinden dönen sonucu almak için aşağıda yer alan method kullanılabilir.

```
private VPosTransactionResponseContract GetResponse()
{
    string msg = "";
    foreach (var item in Request.Form)
        msg += msg;
    WriteLog(msg);

    VPosTransactionResponseContract model = new VPosTransactionResponseContract();
    model.ResponseCode = RemoveDiacritics(Request.Form["ResponseCode"].ToString());
    model.ResponseMessage = RemoveDiacritics(Request.Form["ResponseMessage"].ToString());
    model.ProvisionNumber = Request.Form["ProvisionNumber"].ToString();
    model.Stan = Request.Form["Stan"];
    model.RRN = Request.Form["RRN"];
    model.MerchantOrderId = Request.Form["MerchantOrderId"].ToString();
    string orderId = Request.Form["OrderId"];
    if (!string.IsNullOrWhiteSpace(orderId))
        model.OrderId = Convert.ToInt32(orderId);
    else
        model.OrderId = 0;
    model.HashData = Request.Form["HashData"].ToString();
    model.MD = Request.Form["MD"].ToString();
    return model;
}
```

```
public string RemoveDiacritics(string text)
{
    text = text.Replace("Ä\u0009f", "ğ").Replace("Ä±", "ı");
    return text;
}
```

## 4.6. Cevap Kodlarının Okutulduğu Alanlar

Cevap kodlarının okutulduğu alanlar Asp.net, Klasik Asp, C# ve PHP için aşağıda yer almaktadır.

### Asp.net için;

```
Request.Form["ResponseMessage"].ToString();
Request.Form["ResponseCode"].ToString();
```

### Klasik Asp için;

```
response.write(request.form("ResponseMessage"))
response.write(" " & request.form("ResponseCode"))
```

### C# için;

```
Public ActionResult Success()
{
String ResponseCode=Request.Form["ResponseCode"].ToString();
String ResponseMesaage=Request.Form["ResponseMessage"].ToString();
}

Public ActionResult Fail()
{
String ResponseCode=Request.Form["ResponseCode"].ToString();
String ResponseMesaage=Request.Form["ResponseMessage"].ToString();
}
```

### 4.6.1. Cevap Kodlarının Açıklamaları

#### Cevap Kodu Hata Mesajı

0	OTORİZASYON VERİLDİ
1	KARTI VEREN BANKAYI ARA - LİM.
2	KARTI VEREN BANKAYI ARAYINIZ.
3	GEÇERSİZ ÜYE İŞYERİ
5	İŞLEM ONAYLANMADI

- |    |                                |
|----|--------------------------------|
| 12 | TEKRAR DENEYİNİZ               |
| 11 | VİP-İşlem için onay verildi    |
| 12 | Geçersiz İşlem                 |
| 13 | GEÇERSİZ İŞLEM TUTARI          |
| 14 | Geçersiz Kart Numarası         |
| 15 | Kart Veren Banka Tanımsız      |
| 51 | Bakiyesi-Kredi limiti Yetersiz |
| 53 | DÖVİZ HESABI BULUNAMADI        |
| 54 | Vade Sonu Geçmiş Kart          |
| 55 | Hatalı Kart Şifresi            |
| 53 | Kart Tanımlı Değil.            |
| 57 | İŞLEM TİPİNE İZİN YOK.         |
| 58 | İşlem Tipi Terminale Kapalı    |
| 59 | Sahtekarlık Şüphesi            |
| 62 | KISITLANMIS KART.              |
| 67 | BU HESAPTA HİÇBİR İŞLEM YAPILA |
| 68 | TANIMSIZ ŞUBE                  |
| 75 | Şifre Deneme Sayısı Aşıldı     |
| 76 | Şifreler Uyuşmuyor-Key         |
| 96 | ŞİFRE GÜVENİLİR BULUNMADI      |
| 96 | ARQC KONTROLÜ BAŞARISIZ        |
| 88 | İŞLEM ŞÜPHELİ TAMAMLANDI.KONTR |
| 89 | EK KART İLE BU İŞLEM YAPILAMAZ |
| 90 | Gün Sonu Devam Ediyor          |
| 91 | Kartı Veren Banka Hizmetdişi   |
| 92 | KART VEREN BANKA TANIMLI DEGİL |
| 5  | CVV2 Hatalı                    |
| 96 | İPTAL GERÇEKLEŞTİ              |
| 96 | İPTAL GERÇEKLEŞMEDİ            |

96	İPTAL PROVİZYON
96	KISMİ İPTAL PROVİZYON
96	GEÇERSİZ TAKSİT
96	GEÇERSİZ TAKSİT VADE
96	UYGUN TAKSİT YOK
96	Kart İşyeri Uyumsuz
96	UYGUN FAİZLİ TAKSİT YOK
12	Org işlem bulunamadı
12	Org işlemin tamamı iade edildi
12	iade tutarı kalandan fazla
12	iade tutarı Org Tutar değil
12	Önceden kısmi iade yapılmış
96	Satış Limit Yetersiz
96	Inst Limit Yetersiz
96	Gross Limit 0 veya (-)
96	Installment Limit 0 veya (-)
96	Ek veya Inst Limit Yetersiz
96	Nakit Avans Limit Yetersiz
5	3D CAVV Hatalı
5	3D CAVV Yok
5	3D CAVV ACS Error
57	3D işyerinden işleme izin yok
57	3D Secure Merchant Only