



# HALK ÖDEME ENTEGRASYON TEKNİK DÖKÜMANI

HALK ELEKTRONİK PARA VE ÖDEME HİZMETLERİ VE A.Ş

İSTANBUL



## İÇİNDEKİLER

İÇİNDEKİLER.....	1
1. Giriş .....	2
1.1. Amaç .....	2
1.2. Kapsam.....	2
1.3. Erişim URL'leri .....	2
1.4. Hız Sınırlayıcı .....	2
2. Güvenli İşlem Tipi 2D/3D Kullanımı.....	2
2. 1. Check Order Status.....	7
3. Get Token.....	9
4. Get Installments .....	10
5. Eklentiler .....	12
5.1. WooCommerce Plugin .....	12
5.2. Opencart Module .....	13
5.3. Presta Module .....	14
6. Transaction Refund .....	18
6.1.Refund WebHook.....	20
6.2.Hash anahtarı kullanarak WebHook Yanıt Doğrulamasını iadesi.....	20
7.Recurring WebHook.....	21
7.1. Update Recurring .....	23
7.2. Query Recurring Information .....	25
7.3. Delete Recurring .....	26
7.4. Sale WebHook.....	27
7.5. Merchant Active Installments .....	29
7.6. Pay With White Label Smart 3D .....	30
7.7. Merchant Commission API.....	34
7.8. Hash Anahtarı Kullanarak Response Validation .....	35
7.9. Pay With White Label Smart 2D .....	36
7.10. Pay With CardToken.....	40
7.11. SaveCard Api .....	42
7.12. Save Cart Edit Api .....	44
7.13. SaveCard Delete Api.....	45
7.14. SaveCard : Get Card Tokens Api.....	47
7.15. Get Transactions .....	48
7.16. Payment Confirmation API.....	52
7.17. Pay With Card Token (Non-Secure) .....	53
7.18. All Transaction API .....	57
7.19. Merchant 3D Model : CompletePayment Api.....	62
8.Server Durum Kodları.....	65
9.Uygulama Durum Kodları.....	65



## 1. Giriş

Bu belge, Halkode Ödeme Entegrasyonunun kavramsal bir görünümünü sunmanın yanı sıra üye işyerlerini ve geliştiricilerin ödeme sisteminin çalışma mekanizmasını daha iyi anlamalarına yardımcı olmak için hazırlanmıştır.

### 1.1. Amaç

Bu belgenin amacı, çeşitli ödeme seçenekleriyle ödeme işlemlerini gerçekleştirmek için Halkode Ödeme sistemini web sitelerine entegre etmek üzere üye işyerlerine rehberlik etmektir.

### 1.2. Kapsam

Satıcı uygulamaları ile Halkode ödeme sistemi arasındaki entegrasyonun kapsamı, ödeme entegrasyonu gereksinimi için işlevsel gereksinimi kapsar.

### 1.3. Erişim URL'leri

Ortama bağlı olarak aşağıdaki erişim URL'leri kullanılacaktır.

Ortam	URL Format
Test Ortamı	<a href="https://testapp.halkode.com.tr/ccpayment">https://testapp.halkode.com.tr/ccpayment</a>
Canlı Ortam	<a href="https://app.halkode.com.tr/ccpayment">https://app.halkode.com.tr/ccpayment</a>

### 1.4. Hız Sınırlayıcı

Çift tıklamayı önlemek için tüm Happy Flow API'leri 2 saniye hız sınırlayıcı ile uygulanmıştır, yani 2 saniye içinde arka arkaya aynı birden fazla istek bir istisnaya neden olacaktır.

## 2. Güvenli İşlem Tipi 2D/3D Kullanımı

Halkode yöneticisi, satıcı web sitesinden ödemenin 2D veya 3D kullanılarak yapılıp yapılmayacağına karar verecektir. Satıcı web sitesinden Halkode ödeme entegrasyon sistemine aşağıdaki istek gönderilmelidir. CURL post isteği Halkode'de aşağıdaki parametreler alınmalıdır.

### Request

Method	URL	İçerik-Türü
POST	<ACCESS_URL>/purchase/link	application/x-www-form-urlencoded



Tür	Parametre	Veri Türü	Koşul
KEY	merchant_key	string	Zorunlu
KEY	currency_code	string	Zorunlu
KEY	invoice	string	Zorunlu
KEY	name	string	Zorunlu
KEY	surname	string	Zorunlu
KEY	bill_address1	string	Opsiyonel
KEY	bill_address2	string	Opsiyonel
KEY	bill_city	string	Opsiyonel
KEY	bill_postcode	string	Opsiyonel
KEY	bill_state	string	Opsiyonel
KEY	bill_country	string	Opsiyonel
KEY	bill_email	string	Opsiyonel
KEY	bill_phone	string	Opsiyonel
KEY	sale_web_hook_key	string	Opsiyonel
KEY	ip	string	Opsiyonel
KEY	transaction_type	string	Opsiyonel
KEY	vpos_type	string	Opsiyonel
KEY	identity_number	string	Opsiyonel
KEY	is_cvv_less	string	Opsiyonel
KEY	installments_number	string	Opsiyonel

#### Request for Recurring

KEY	order_type	<i>integer</i>	Zorunlu
KEY	recurring_payment_number	<i>integer</i>	Zorunlu
KEY	recurring_payment_cycle	<i>string</i>	Zorunlu
KEY	recurring_payment_interval	<i>integer</i>	Zorunlu
KEY	recurring_web_hook_key	<i>string</i>	Zorunlu

**merchant\_key** : Satıcıya atanan benzersiz bir anahtardır. Üye İşyerinin web sitesinden gönderilmelidir.

**invoice**: Ürün adı, miktarı ve birim fiyatı vb. listesiyle birleştirilmiş json formatında bir dizidir.

Örneğin, üç ürün varsa,

Ürün 1#

Name: Item1, Qty: 2, Unit Price: 200

Ürün 2#

Name: Item2, Qty: 1, Unit Price: 100

Ürün 3#

Name: Item3, Qty: 2, Unit Price: 400

#### Fatura bilgisi aşağıdaki gibi json dizi olacaktır

\$invoice['invoice\_id'] = "345345535"; // *İade veya iptal URL'sinde bulunacak benzersiz bir id*

\$invoice['invoice\_description'] = "*Fatura Test Açıklaması*";

\$invoice['total'] = 1300

\$invoice[discount] = 220 // *Kupon kodu veya indirim değeri tutarı*

\$invoice[coupon] = "3XY8P" // *geçerli olması durumunda kupon kodu*



```
$invoice['return_url'] = "https://<your_success_url>"
$invoice['cancel_url'] = "https://<your_fail_or_cancel_url>"
$invoice['items'] = array(
    array("name"=>"Item1","price"=>200,"quantity"=>2,"description"=>"item1 description"),
    array("name"=>"Item2","price"=>100,"quantity"=>1,"description"=>"item2 description"),
    array("name"=>"Item3","price"=>400,"quantity"=>2,"description"=>"item3 description"),
);
//fatura bilgisi
$invoice['bill_address1'] = 'Address 1'; // 100 karakterden fazla olmamalıdır
$invoice['bill_address2'] = 'Address 2'; // 100 karakterden fazla olmamalıdır
$invoice['bill_city'] = 'Istanbul';
$invoice['bill_postcode'] = '1111';
$invoice['bill_state'] = 'Istanbul';
$invoice['bill_country'] = 'TURKEY';
$invoice['bill_phone'] = '008801777711111';
$invoice['bill_email'] = 'demo@halkode.com.tr';
$invoice['sale_web_hook_key'] = 'sale_web_hook_key'; //Bu anahtar Halkode Üye İşyeri Panelinde atanmalıdır.
$invoice['transaction_type'] = 'Pre-Authorization'; //İsteğe bağlı zorunlu değil
$invoice['vpos_type'] = 'insurance': Yalnızca Pay2D çözümünün bir sigorta Satıcısı tarafından bir sigorta ödemesi başlatması için geçerlidir. Sunulan değer "sigorta" olmalıdır, aksi takdirde bir doğrulama istisnası atacaktır.
$invoice['identity_number'] = '12345678901': Yalnızca Pay2d Çözümü için "vpos_type": "insurance" mevcut olduğunda bir sigorta Satıcısı tarafından bir sigorta ödemesi başlatmak için geçerlidir. Değer (tckn/vkn/tin) 10-11 hane arasında olmalıdır.
//Yinelenen bilgi
$invoice['order_type'] = 1; //yinelenen ödeme için sipariş tipi 1
$invoice['recurring_payment_number'] = 2; //tamsayı bir değer olmalıdır.
$invoice['recurring_payment_cycle'] = 'M'; // örnek: D, M, Y
$invoice['recurring_payment_interval'] = 2; // tamsayı bir değer olmalıdır.
$invoice['recurring_web_hook_key'] = 'key_name'; //Bu anahtar Halkode üye işyeri panelinde atanmalıdır.

Tüm vergiler ve kargo ücretleri fatura kalemleri dizisine kalem olarak eklenecektir. Öğe adları, miktar 1 olacak şekilde sırasıyla "Vergi" ve "Nakliye Ücreti" olmalıdır.
```

#### Nihayi Request Deseni:

```
$invoice = json_encode($invoice);
$post = array(
    'merchant_key' => $merchant_key,
    'invoice' => $invoice,
    'currency_code' => $currency,
    'name' => $name,
    'surname' => $surname
);
```

**currency\_code:** Para biriminin kodudur. Örneğin USD, TRY, EUR vb.

**Name: Adı:** Adı alanı örneğin, ürünü satın alan kişinin adı "John Dao" ise, adı "John" olmalıdır

**Surname:** Soyadı alanı. Örneğin, ürünü satın alan kişinin adı "John Dao" ise, soyadı "Dao" olmalıdır.



**sale\_web\_hook\_key:** İsteğe bağlı bir anahtardır. Bir satın alma talebi tamamlandığında, Halkode bir post isteği gönderir. Böylece satıcı kendi sitesinde bir etkinlik gerçekleştirebilir. Halkode, bu anahtarın veritabanında var olması gerektiğini doğrular. Satıcı, Halkode Satıcı Panelinde bu anahtara karşı bir Satış web kancası URL'si atamalıdır.

**transaction\_type:** "PreAuth" için sistemde bekleyen bir işlem oluşturulur. Ancak daha sonra üye işyeri onayı ile Tamamlandı'ya dönüştürülür.

**order\_type:** Order\_type=1 ise, Halkode ödemeyi yineleme için doğrular. Bu durumda; recurring\_payment\_number, recurring\_payment\_cycle, recurring\_payment\_intervalkeys boş olmamalıdır.

**recurring\_payment\_number:** Taksit sayısını tanımlar. Eğer first\_amount 100\$ ise ve recurring\_payment\_number 5 ise, toplam tutar  $\$100 \times 5 = \$500$  olmalı. (İşlem maliyeti her işlemle birlikte eklenebilir)

**recurring\_payment\_cycle:** Ödeme aralığı zamanı tanımlanır. Alacağı değerler: D /M/Y olup D: Gün, M: Ay, Y: Yıl ifade eder.

**recurring\_payment\_interval:** Ödeme aralık süresi tanımlanır. Eğer recurring\_payment\_interval= 2 ve recurring\_payment\_cycle = "M" ise işlem her 2 ayda bir gerçekleşecektir.

**recurring\_web\_hook\_key:** Satıcı yinelenen web kancası url'sini tanımlar. Halkode Üye İşyeri Panelinde bu anahtara karşılık bir URL atanmalıdır. Halkode, bu anahtarın veritabanında bulunması gerektiğini doğrular ve ödeme tekrarlandığında gerekli bir değerdir.

**is\_cvv\_less :** Eksik ödeme varsa 1 yoksa 1 gönderilir.

**installments\_number:** Kartlı ödeme sayfasında taksit göstermek veya seçmek istemiyorsanız taksit numarasını istekte gönderin

#### **Response:**

Başarılı bir istekten sonra, sunucu CURL göndericisine bir bağlantı sağlayacaktır. Satıcı web sitesinin ödeme yapmak için siteye yönlendirilmesi gerekir.

Yanıta aşağıdaki anahtarlar olacaktır:

KEY	status	string
KEY	success_message	string
KEY	link	string

**status:** API isteğinin sonucudur. Başarı için "true" ve başarısızlık için "false".

**success\_message:** İsteğin durumunu açıklayan ifadedir.

**link:** Satıcı web sitesinin 2d/3d ödeme işlemleri için yönlendirilmesi gereken URL'dir.

**Response Redirect:** Ödeme işleminden sonra, son kullanıcı aşağıda verilen parametreler kullanılarak satıcı web sitesine yönlendirilir. Yanıt doğrulaması için bir hash\_key parametresi gönderiyoruz. Satıcı, istek parametresini 7.8 de açıklanan hash\_key şifresi çözülmüş değerle karşılaştırabilir.



```
$params = [  
  'payment_status' => $status,  
  'order_no' => $order_id,  
  'invoice_id' => invoice_id,  
  'status_code' => $error_code,  
  'status_description' => $status_message,  
  'payment_method' => $payment_type,  
  'transaction_type' => $transaction_type,  
  'error_code' => $error_code,  
  'error' => $status_message,  
  'hash_key' => $hash_key,  
];
```

**payment\_status** : 0=Failed, 1=Completed

**order\_id**: Halkode sipariş id.

**invoice\_id**: Üye işyeri fatura id.

#### Response Parametre Açıklaması

Anahtar	Açıklama
payment_status	1/0 olabilir. 1=başarı, 0=başarısız
order_no	Halkode sipariş no
invoice_id	Üye işyeri fatura id
status_code	Halkode Durum kodu, 100 başarı kodudur
status_description	İşlem açıklaması
payment_method	1= Kredi Kartı, 2= Mobil, 3= Cüzdan
transaction_type	transaction_type == "Auth" //işlem tutarı karttan anında düşülür. transaction_type == "Pre-Authorization" //işlem tutarı karttan daha sonra düşülecektir.
error_code	Status_code parametresinin değeri
Error	status_description parametresinin değeri
hash_key	Talebin Halkode'dan geldiğini doğrulamak için

#### Yanıtı Doğrulama:

**Koşul 1** : payment\_status == 1 and transaction\_type == "Auth" // İşlem başarılıdır ve işlem tutarı karttan anında düşülür.

**Koşul 2** : payment\_status == 1 and transaction\_type == "Pre-Authorization" //işlem başarılıdır ve işlem tutarı karttan daha sonra çekilecektir.

**Koşul 2**: payment\_status == 0 //işlem başarısız

hash\_key, talebin Halkode'dan geldiğini doğrulamak için satıcı tarafında doğrulanmalıdır. Ayrıca Sipariş Durumu API'si (2.1) işlemin başarılı veya Başarısız olduğunu onaylar.



## 2. 1. Check Order Status

Ödeme 2D/3D kullanılarak başarıyla tamamlandığında, Halkode ödeme sistemi kullanıcıları satıcının başarı URL'sine yönlendirir. Satıcı web sitesinde sipariş durumu "Tamamlandı" olarak değiştirilmeli ve sepet temizlenmelidir. Halkode'dan gerçek sipariş durumunu öğrenmek için, Halkode'ye fatura kimliği ve satıcı anahtarı ile gönderilen başka bir istek olmalıdır. getOrderStatus API'sinin isteği ve yanıtı:

### Request

Method	URL	Content-Type
POST	<ACCESS_URL>/api/checkstatus	application/json

Tür	Parametre	Veri Türü	Koşul
HEADER	Authorization	string	Zorunlu
HEADER	Accept	string	Zorunlu
KEY	invoice_id	string	Zorunlu
KEY	merchant_key	string	Zorunlu
KEY	hash_key	string	Zorunlu

### Yetkilendirme

Yetkilendirme, bağlantı girişimine izin verildiğine dair doğrulamayı tanımlayan bir başlık anahtarıdır. Yöntem "Bearer" olmalıdır. Örnek değer:

#### Bearer

eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFjZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFIMDZkZT

**Accept:** İstemci tarafında ne tür bir gösterim istendiğini belirler. Değer "application/json" olmalıdır

**invoice\_id:** Satıcı tarafından gönderilen benzersiz bir sipariş kimliğidir

**merchant\_key:** Halkode tarafından sağlanan satıcının benzersiz anahtarıdır.

**hash\_key:** Hash\_key oluşturma yöntemi aşağıda verilmiştir.

```
function generateRefundHashKey($invoice_id, $merchant_key, $app_secret) {
    $data = $invoice_id.'|'.$merchant_key;
    $iv = substr(sha1(mt_rand()), 0, 16);
    $password = sha1($app_secret);
    $salt = substr(sha1(mt_rand()), 0, 4);
    $saltWithPassword = hash('sha256', $password . $salt);
    $encrypted = openssl_encrypt(
        "$data", 'aes-256-cbc', "$saltWithPassword", null, $iv
    );
    $msg_encrypted_bundle = "$iv:$salt:$encrypted";
    $hash_key = str_replace('/', '_', $msg_encrypted_bundle);
    return $hash_key;
}
```

### Başarısız Response:

```
{
    "status_code": 41,
    "status_description": "Order Failed",
    "transaction_status": "Failed",
}
```





```
"order_id": "162435924998223",
"transaction_id": "6jvQ7-l3kb-TC10-98950-220621",
"message": "Order Failed",
"reason": "Failed # transaction failed",
"bank_status_code": "Failed # transaction failed",
"bank_status_description": "transaction failed",
"invoice_id": "50781624359247",
"total_refunded_amount": 0,
"product_price": "20.00",
"transaction_amount": 21.2,
"ref_number": "",
"transaction_type": "Pre-Authorization",
"merchant_commission": "2.18"
"user_commission": "0.96",
"md_status": 4,
"original_bank_error_code": "43",
"original_bank_error_description": "Çalıntı Kart-Karta el koyunuz",
}
```

#### **Başarılı Response:**

```
{
  "status_code": 100,
  "status_description": "An order has been taken place for this invoice id: 33491162435928",
  "transaction_status": "Completed",
  "order_id": "162435932934307",
  "transaction_id": "SXY0m-o3kb-TC10-98950-220621",
  "message": "An order has been taken place for this invoice id: 33491162435928",
  "reason": "",
  "bank_status_code": "",
  "bank_status_description": "",
  "invoice_id": "33491162435928",
  "total_refunded_amount": 0,
  "product_price": "20.00",
  "transaction_amount": 22.1,
  "ref_number": ""
  "transaction_type": "Auth",
  "merchant_commission": "2.68"
  "user_commission": "1.06",
  "md_status": 1,
  "settlement_date": "2022-04-22",
  "original_bank_error_code": "",
  "original_bank_error_description": "",
}
```



### Başarılı Response for Recurring:

```
{
  "status_code": 100,
  "status_description": "An order has been taken place for this invoice id: 16014922137077",
  "transaction_status": "Completed",
  "order_id": "16014922157187",
  "transaction_id": "iH8MK-EQe-TC10-98950-300920",
  "message": "An order has been taken place for this invoice id: 16014922137077",
  "reason": "",
  "bank_status_code": "",
  "bank_status_description": "",
  "invoice_id": "16014922137077",
  "total_refunded_amount": 0,
  "product_price": "100.00",
  "transaction_amount": 100,
  "recurring_id": 303,
  "ref_number": "",
  "recurring_plan_code": "1601492241FdsraX",
  "next_action_date": "2021-05-30 03:10:00",
  "recurring_status": "Active",
  "transaction_type": "Auth",
  "merchant_commission": "7.68",
  "user_commission": "4.06",
}
```

### 3. Get Token

API, satıcıları doğrulamak için diğer API'lerde kullanılmak üzere bir token oluşturur. Ayrıca satıcı için ayarlanmış bir ödeme entegrasyon seçeneği de döndürür. Yanıt anahtarı "is\_3d" dir. is\_3d'nin olası değerleri şunlardır: 0, 1, 2 ve 4.

0 = Yalnızca beyaz etiketli 2D, 1 = Beyaz etiketli 2D veya 3D, 2 = Yalnızca beyaz etiketli 3D, 4 = Markalı ödeme çözümü

Belirteç Api'si 1 döndürürse, satıcı web sitesi kullanıcının 2D veya 3D'yi seçmesi için bir onay kutusu görüntülemelidir.

**Her tokenin 2 saatlik son kullanma süresi vardır ve süresi dolana kadar aynı tokenin kullanılması şiddetle tavsiye edilir. Bir satıcı için bir token geçerlilik süresi kapsamında devam etmek için her istek için bu API'yi her seferinde çağırmaya gerek yoktur.**

Aşağıdaki istek ve yanıt API'nin örnekleridir.

#### Request

Metot	URL	İçerik Türü
POST	<ACCESS_URL>/api/token	application/json

Tür	Parametre	Veri Türü	Koşul
KEY	app_id	string	Zorunlu
KEY	app_secret	string	Zorunlu

**app\_id:** Satıcıya atanan benzersiz bir anahtardır. Bir satıcı web sitesinden gönderilmelidir.

**app\_secret:** Satıcıya verilen benzersiz ve gizli bir anahtardır. Bir satıcı web sitesinden gönderilmelidir.

**Başarısız Response:**

```
{
  "status_code": 2,
  "status_description": "Invalid credentials"
}
```

**Başarılı Response:**

```
{
  "status_code": 100,
  "data": {
    "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtp0aSI6IjEwNzlkZDMxZDlhOWJmMGZiNmI3NGIj",
    "is_3d": 1
  },
  "status_description": "Successfully Generated token"
"expires_at": "2021-01-18 20:20:10"
}
```

## 4. Get Installments

API, kullanıcı kart numarasını girdiğinde ve uzunluğu altı haneyi aştığında çağrılmalıdır. getPos API, ödeme sayfasında verilen kart numarasına göre taksitlerin bir listesini sağlamaktan sorumludur.

### Request

Metot	URL	İçerik Türü
POST	<ACCESS_URL>/api/getpos	application/json

Tür	Parametre	Veri Türü	Koşul
HEADER	Authorization	string	Zorunlu
HEADER	Accept	string	Zorunlu
KEY	credit_card	digit	Zorunlu
KEY	amount	double	Zorunlu
KEY	currency_code	string	Zorunlu
KEY	merchant_key	string	Zorunlu
KEY	is_recurring	int	Opsiyonel

## Yetkilendirme

Yetkilendirme, bağlantı girişimine izin verildiğine dair doğrulamayı tanımlayan bir başlık anahtarıdır. Yöntem "Bearer" olmalıdır. Örnek değer:

## Bearer

eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFjZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzYlMDZkZT  
vZjph9ifRB85TGwW4HimQk46RPG9Hp6kydLnuhFOkbvGpaxcs5qyZ67-

**accept** : Kabul için istemci tarafında ne tür bir gösterim istendiğini belirler. Değer "application/json" olmalıdır.

**credit\_card:** Kredi kartı için istemci tarafında ne tür bir gösterim istendiğini belirler. Değer "application/json" olmalıdır. **Kredi kartı numarasının tamamı gönderilmelidir.**

**amount:** Toplam ürün miktarıdır



**currency\_code:** Para biriminin ISO kodudur. Örneğin, USD, TRY, EUR vb.

**merchant\_key:** Halkode tarafından sağlanan satıcının benzersiz anahtarıdır.

**is\_recurring:** Ödeme yinelenenirse 1 olmalı.

**is\_2d:** Get token API yanıtında, "is\_3d" 0 ise is\_2d=1 gönderilmelidir;

**Başarısız Response:**

```
{
  "status_code": 3,
  "status_description": "merchant not found"
}
```

**Başarılı Response:**

```
{
  "status_code": 100,
  "status_description": "Successful",
  "data": [
    {
      "pos_id": 1,
      "campaign_id": 0,
      "allocation_id": 0,
      "installments_number": 1,
      "amount_to_be_paid": 23,
      "currency_code": "TRY",
      "currency_id": 1,
      "title": "Single payment",
      "hash_key": "5uUVKijz5im5FfStic2wVX4gG8ngRfMS3H+FvAauQvOc1nAnh9GZ9T6zyxV5WUnQP2F'
    }
  ],
}
```

**Başarılı response tarım kredi kartı veya bin numarası**

Sağlanan kredi kartı veya kutu numarası Tarım/Ziraat ise başarı yanıtı şu şekildedir

**Başarılı Response:**

```
{
  "status_code": 100,
  "status_description": "Successful",
  "data": [
    {
      "pos_id": 1,
      "campaign_id": 0,
      "allocation_id": 0,
      "installments_number": 1,
      "amount_to_be_paid": 23,
      "currency_code": "TRY",
      "currency_id": 1,
      "title": "Single payment",
      "hash_key": "5uUVKijz5im5FfStic2wVX4gG8ngRfMS3H+FvAauQvOc1nAnh9GZ9T6zyxV5WUnQP2F',
      "maturity_period": 1,
      "payment_frequency": 1
    }
  ]
}
```



},  
}

**Not:** Veri değerleri bir dizi olabilir. Bu durumda, üye işeri web sitesi kullanıcının seçmesi için tüm taksit seçeneklerini göstermelidir. Varsayılan olarak, ilk taksit seçilecektir. Her durumda, yanıtta en az bir taksit mevcut olacaktır.

Müşteri talebine dayanarak, belirli bir marka için yanıt olarak card\_issuer\_name eklenmiştir.

## 5. Eklentiler

Aşağıdaki eklentiler, Halkode ödeme sistemini entegre etmek için özel kod yazmadan satıcı web sitesine yüklenebilir.

### 5.1. WooCommerce Plugin

WooCommerce eklentisini yüklemek için aşağıdaki adımları izlenir.

Adım#1: Go to plugins=>add plugin

Adım#2: browse the plugin zip file and activate.

Adım#3: Başarılı bir aktivasyondan sonra, eklentiği yapılandırmak için aşağıdaki ekran görünmelidir.

Pages  
Comments  
Projects  
WooCommerce  
Orders  
Coupons  
Reports  
Settings  
Status  
Extensions  
Products  
WPForms  
Appearance  
Plugins 3  
Users  
Tools  
All-in-One WP Migration  
Settings  
Divi  
LiteSpeed Cache  
Collapse menu

Merchant Key  
\$2y\$10\$J6Dyx/KPw3wVa38NOcDxT.Wf8Dg1Od9UkX4siWmOtC  
Get your secure secret from Sipay

Merchant ID  
56  
Get your Merchant Id from Sipay

Merchant Currency Code  
TRY

Merchant Currency Symbol  
\$

Merchant Request URL  
https://provisioning.sipay.com.tr/ccpayment/purchase/link

Merchant Transaction Status URL  
https://provisioning.sipay.com.tr/request/status

Return URL  
http://www.shareideas.info/success

Cancel URL  
http://www.shareideas.info/fail

Save changes

If you like WooCommerce please leave us a ★★★★★ rating. A huge thanks in advance!

Version 5.2.3

Adım#4: Anahtarı, dönüş url'sini, iptal url'sini vb. yapılandırın ve "Save Changes"e tıklayın. Ödeme sayfasında, Halkode ödeme seçeneği aşağıdaki gibi görünmelidir:

Your order

Product	Total
The Witcher: Wild Hunt Collector's Edition - PlayStation 4 ×2	₺8,000.00
Subtotal	₺8,000.00
Total	₺8,000.00

Pay with sipay

Please remit payment to Store Name upon pickup or delivery.









PAY WITH SIPAY

































## 5.2. Opencart Module

1. Upload zip file to your Host.
2. Extract zip file after that logged in your OpenCart 3.x admin
3. Go to extension => ödemeyi seçin ve açılan kutudan HalkÖde'yi arayın ve yüklemek için + düğmesine basın.

The screenshot shows the OpenCart 3.x admin interface. On the left, the 'Extensions' menu is highlighted with a red arrow. In the center, a dropdown menu is open, showing a list of extensions with 'Payments (48)' selected. Below this, a table titled 'Payments' lists various payment methods. A red arrow points to the 'HalkÖde' entry in the table.

Payment Method	Status	Sort Order	Action
Alipay Pay	Disabled		 
Alipay Cross-border	Disabled		 
Amazon Pay and Login with Amazon	Disabled		 
Authorize.Net (AIM)	Disabled		 

This screenshot shows a detailed view of the 'Payments' extension list. A red arrow points to the 'HalkÖde' entry, which is currently 'Enabled'. The table lists various payment methods with their logos, status, sort order, and action buttons (edit and add).

Payment Method	Status	Sort Order	Action
Payza	Disabled		 
Perpetual Payments	Disabled		 
Pilibaba for Chinese Checkout	Disabled		 
PayPal (Powered by Braintree)	Disabled		 
Realex Redirect	Disabled		 
Realex Remote	Disabled		 
SagePay Direct	Disabled		 
SagePay Server	Disabled		 
Sage Payment Solutions (US)	Disabled		 
Secure Trading Payment Pages	Disabled		 
Secure Trading Web Service	Disabled		 
HalkÖde	Enabled		 
Skrill	Disabled		 
Square	Disabled		 
2Checkout	Disabled		 

4. Yüklendiğinde ayarları yapmak için edit düğmesi göreceksiniz. Buna tıklayın ve api domain (prod için <https://app.halkode.com.tr> ve test ortamı için <http://testapp.halkode.com.tr>), app key, app secret, merchantid vb. ayarlayın.



**Edit SIPAY Payment**

\* Merchant ID: 78640

\* Merchant Key: \$2y\$10\$0X.RKmbNjKHg7vfj8N46j.Zq.AU6vBVASro7AGGkaffB4mrdaV4mO

\* Merchant API Domain: https://provisioning.sipay.com.tr

\* Merchant Currency: Your selected currency : USD  
Turkish lira (₺)

\* App ID: 077faac7dba364b3f058193de9fea2e6

\* App Secret: bb18138fbd6fe9a2512e8933e6f37a01

\* Label used at checkout page: Pay now

Total: Total

Order Status: Complete

Geo Zone: All Zones

### 5.3. Presta Module

- Go to Presta Shop's admin page.
- Click on Modules and Services> Modules and Services

**PrestaShop 1.6.1.34** Rajib Kant ID Quick Access My shop System Administrator

**Modules and Services**

**List of modules**

Update all Add a new module Help

Addons membership provides access to all our PrestaShop modules.  
Once connected, your new modules will be automatically installed.

[SIGN UP](#) [LOG IN](#)

Installed Modules: 66 Disabled Modules: 0 Modules to update: 1

An upgrade is available for some of your modules! [Merchant Expertise](#)

**MODULES LIST**

Search:

Filter by: Installed & Not installed Enabled & Disabled Authors: All authors

**Payments and Gateways**  
v1.3.9 - by PrestaShop Partners - [Official](#)  
PayPal is the fast, easy, secure way to accept payments online in 24 currencies from 193 countries around the globe. Your customers don't need a PayPal account to pay and you get paid quickly.  
[Read more](#) [Install](#)

**Payments and Gateways**  
Amazon Pay - Login and Pay with Amazon v2.3.3 - by patwors multimedia GmbH - [Official](#)  
Set up Amazon Pay as a payment method and enable all Amazon users to buy without leaving your Prestashop site.  
[Configure](#)

- "Add a new module " buttonuna tıklayın



PrestaShop 1.6.1.24 Rajib Kant ID 176 Quick Access Connect to PrestaShop Marketplace account My shop System Administrator

Modules and Services

## List of modules

Update all Add a new module Help

Addons membership provides access to all our PrestaShop modules. Once connected, your new modules will be automatically installed. [SIGN UP](#) [LOG IN](#)

Installed Modules 66 Disabled Modules 0 Modules to update 1

An upgrade is available for some of your modules! [Merchant Expertise](#)

### MODULES LIST

Search Filter by Installed & Not Installed Enabled & Disabled Authors All authors

Favorites	0			
All	268			
Administration	32			
Advertising and Marketing	18			

Payments and Gateways  
v1.3.9 - by PrestaShop Partners - [Official](#)  
PayPal is the fast, easy, secure way to accept payments online in 24 currencies from 193 countries around the globe. Your customers don't need a PayPal account to pay, and you get paid quickly  
[Read more](#)

Install

Payments and Gateways  
Amazon Pay - Login and Pay with Amazon v2.3.3 - by patworx multimedia GmbH - [Official](#)  
Set up Amazon Pay as a payment method and enable all Amazon users to buy without leaving your Prestashop site

Configure

"choose a file" buttonuna tıklayın

- Halköde modülünün dosyasını ekleyin.

"Upload this module" tıklayın.

PrestaShop 1.6.1.24 Rajib Kant ID 176 Quick Access Connect to PrestaShop Marketplace account My shop System Administrator

Modules and Services

## List of modules

Update all Add a new module Help

### ADD A NEW MODULE

The module must either be a Zip file (.zip) or a tarball file (.tar, .tar.gz, .tgz).

Module file  Choose a file

Upload this module

### MODULES LIST

Search Filter by Installed & Not Installed Enabled & Disabled Authors All authors

Favorites	0			
All	268			
Administration	32			
Advertising and Marketing	18			

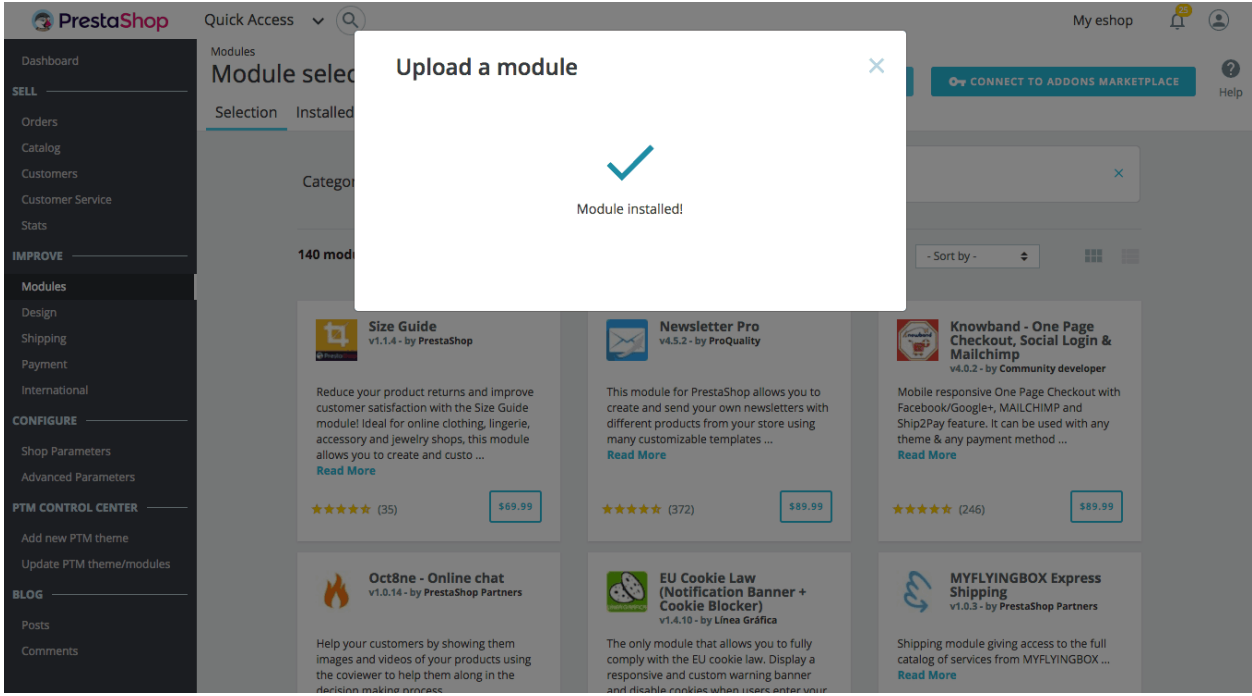
Payments and Gateways  
v1.3.9 - by PrestaShop Partners - [Official](#)  
PayPal is the fast, easy, secure way to accept payments online in 24 currencies from 193 countries around the globe. Your customers don't need a PayPal account to pay, and you get paid quickly  
[Read more](#)

Install

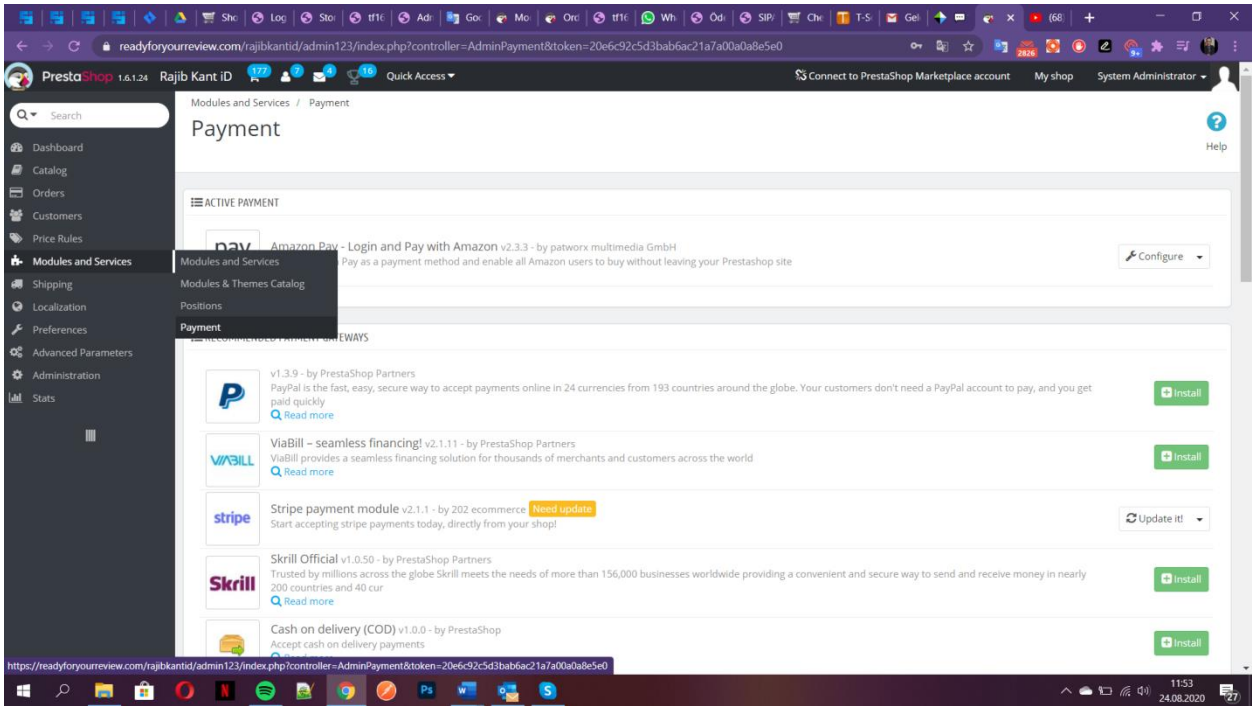
Payments and Gateways  
Amazon Pay - Login and Pay with Amazon v2.3.3 - by patworx multimedia GmbH - [Official](#)  
Set up Amazon Pay as a payment method and enable all Amazon users to buy without leaving your Prestashop site

Configure





- Modules and Services menüsüne tıklayın> Yüklendiğinde ayarları yapmak için Payment linkine tıklayın.



- Click on the configuration next to the order module.



Modules and Services

### List of modules

Payments and Gateways

POS / TPV CECA (Secure payments, refunds, SHA256) v4.2.11 - by OBS Solutions - **Popular** \$59.99

Payments and Gateways

Rock POS - Point of Sale and Omnichannel v4.2.5 - by Hamsa - **Popular** \$359.99

Payments and Gateways

Sipay v1.1.2 - by PrestaShop **Configure**

Payments and Gateways

Sips 3x Worldline Atos v3.1.0 - by PrestaShop - **Popular** \$359.99

Payments and Gateways

Skrill Official v1.0.50 - by PrestaShop Partners - **Official** **Install**

Payments and Gateways

Square Payment v1.3.6 - by Community developer - **Popular** \$179.99

Payments and Gateways

Stripe payment module v2.1.1 - by 202 ecommerce - **Official** **Update it**

Buna tıklayın ve api domain (prod için <https://app.halkode.com.tr> ve test ortamı için <http://testapp.halkode.com.tr>), app key, app secret , merchant id vb. ayarlayın.  
Aşağıdaki kaydet butonu ile kaydedilir.

Modules and Services / sipay / **Configure**

### Configure Sipay

This module allows you to accept secure payments by sipay.

ACCOUNT DETAILS

\* Merchant Key: s2ys10sP2pc.eAQ7Xws4IF2P2ynWu4C4Wk9FI/AITjxvqdyg5GzpjlsMn1Hu

\* Merchant ID: 56307

\* App ID: 2fe2a27fb56c8c195ca97e49a9815a4b

\* App Secret: 68f18a5aa131274aec01fa3c41c1c9b

\* Merchant Currency: Turkish lira (₺)

\* Api domain: https://provisioning.sipay.com.tr

\* Label Of Button: Devam

**Save**



## 6. Transaction Refund

Metot	URL	İçerik Türü
POST	<ACCESS_URL>/api/refund	Application/json

Tür	Parametre	Veri Türü	Koşul
HEADER	Authorization	string	Zorunlu
HEADER	Accept	string	Zorunlu
KEY	invoice_id	string	Zorunlu
KEY	amount	double/empty	Zorunlu
KEY	app_id	string	Zorunlu
KEY	app_secret	string	Zorunlu
KEY	merchant_key	string	Zorunlu
Key	hash_key	string	Zorunlu
Key	refund_transaction_id	string	Zorunlu
key	refund_web_hook_key	string	Opsiyonel

### Authorization

**Authorization** is a header key which defines verification that the connection attempt is allowed. The method should be "Bearer"

Example value:

Bearer

eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFjZjdhdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFIMDZkZT

**Accept:** İstemci tarafında ne tür bir gösterim istendiğini belirler. Değer "application/json" olmalıdır

**invoice\_id:** Satıcı tarafından gönderilen benzersiz bir sipariş kimliğidir

**amount:** İade edilecek tutar

**Not: Tutar boş dize ("" ) olarak ayarlanırsa, bir işlemin tutarının geri kalanı iade edilecektir.**

**Burada, tutarın geri kalanı = bir işlemin toplam tutarı - kısmi iadelerin toplam tutarı**

**app\_id**

**app\_secret:** Halkode tarafından sağlanan benzersiz kimliktir

**merchant\_key:** Halkode tarafından sağlanan satıcının benzersiz anahtarıdır.



## Hash Key Generation Method

```
function generateRefundHashKey($amount, $invoice_id, $merchant_key, $app_secret) {  
  
    $data = $amount.'|'.$invoice_id.'|'.$merchant_key;  
  
    $iv = substr(sha1(mt_rand()), 0, 16);  
    $password = sha1($app_secret);  
  
    $salt = substr(sha1(mt_rand()), 0, 4);  
    $saltWithPassword = hash('sha256', $password . $salt);  
  
    $encrypted = openssl_encrypt(  
        "$data", 'aes-256-cbc', "$saltWithPassword", null, $iv  
    );  
  
    $msg_encrypted_bundle = "$iv:$salt:$encrypted";  
    $hash_key = str_replace('/', '___', $msg_encrypted_bundle);  
  
    return $hash_key;  
}
```

### **Başarısız Request:**

```
{  
    "status_code": 49,  
    "status_description": "Refund Failed",  
    "order_no": "15925741639038",  
    "invoice_id": "66955",  
    "ref_no": ""  
}
```

### **Başarılı Request:**

```
{  
    "status_code": 100,  
    "status_description": "Refund completed successfully",  
    "order_no": "15925741639038",  
    "invoice_id": "66955",  
    "ref_no": "5454545dgdgd545545"  
}
```



## 6.1.Refund WebHook

Satıcının <https://app.halkode.com.tr/merchant/apisetting> adresindeki Halkode satıcı panelinde iade web kancası URL'sini (key, value) ayarlaması gerekir. Geri ödeme için, refund\_web\_hook\_key anahtarı geri ödeme isteği parametresiyle birlikte gönderilmelidir. Halkode iade talebi yaparken anahtarın veritabanında var olduğunu doğrular.

Geri ödeme onayında, aşağıda verilen parametrelerle bir satıcı geri ödeme web kancası url'sine bir POST isteği gönderilir.

Tür	Parametere	Örnek Değer
Key	invoice_id	8iu75g
Key	order_id	15767887576675
Key	amount	10.50
Key	status	Completed
Key	hash_key	3rter3dtrytyutuer5y5tyttryrtryry=

## 6.2.Hash anahtarı kullanarak WebHook Yanıt Doğrulamasını iadesi

Geri Ödeme Onayında, Halkode web kancası url'sine bir gönderi isteği gönderir. Bu bağlantılarla ilgili sorun, anonim bir kişi tarafından erişilebilmeleridir. Bu sorunu önlemek için, istek parametreleri ile durum, fatura\_id, sipariş\_id, tutar ve hash\_key gibi bağlantılara bazı parametreler eklendiğinden, isteğin hash anahtarı kullanılarak doğrulanması önerilir.

```
function validateHashKey($hash_key, $secret_key){  
    $status = "";  
    $amount = $invoice_id = $order_id = 0;  
    if (!empty($hash_key)) {  
        $hash_key = str_replace('_', '/', $hash_key);  
        $password = sha1($secret_key);  
        $components = explode(':', $hash_key);  
        if (count($components) > 2) {  
            $iv = $components[0] ?? "";  
            $salt = $components[1] ?? "";  
            $salt = hash('sha256', $password . $salt);  
            $encrypted_msg = $components[2] ?? "";  
            $decrypted_msg = openssl_decrypt($encrypted_msg, 'aes-256-cbc', $salt, null, $iv);  
            if (strpos($decrypted_msg, '|') !== false){  
                $array = explode('|', $decrypted_msg);  
            }  
        }  
    }  
}
```



```
$status = $array[0] ?? 0;

$amount = $array[1] ?? 0;

$invoice_id = $array[2] ?? '0';

$order_id = $array[3] ?? 0;

}

}

}

return [$status, $amount, $invoice_id, $order_id];

}
```

Burada, \$hash\_key bir istekten alınmalıdır ve \$secret\_key Halkode'dan sağlanan bir satıcı uygulama sırrıdır.

## 7.Recurring WebHook

Öncelikle, <https://app.halkode.com.tr/merchant/apisetting> adresindeki Halkode satıcı panelinde yinelenen web kancası URL'nizi (key, value) ayarlamanız gerekir. Yinelenen ödeme için, satıcıların *recurring\_web\_hook\_keywith* invoice ayarlaması gerekir. Halkode, bir satın alma talebi oluştururken bu anahtarın veritabanında var olduğunu doğrular.

Her yinelenen ödeme için SHalkode, aşağıda verilen parametrelerle birlikte bir satıcı yinelenen web kancası url'sine bir POST isteği gönderir.

Tür	Parametre	Örnek Değer
Key	merchant_key	\$2hiu3er3iejdfjkvi343hh553h34k34h
Key	invoice_id	8iu75g
Key	order_id	15767887576675
Key	product_price	10.50
Key	plan_code	15767Ythgyuy
Key	recurring_number	2
Key	status	"Completed"/ "Failed"
Key	attempts	10
Key	action_date	20-08-19 14:15:48
Key	hash_key	Ssfs:sdfsfs:4rfdgdgddfdgdgdg=



## WebHook uygulama için kılavuz:

**Adım 1:** Doğrulama isteği POST'tur ve satıcı anahtarı geçerlidir.

**Adım 2:** Bundan sonra, yinelenen sorgu API'sini çağırın.

Metot	URL	İçerik Türü
POST	<ACCESS_URL>/api/recurringPlan/query	application/json

Tür	Parametre	Veri Türü	Koşul
HEADER	Authorization	string	Zorunlu
HEADER	Accept	string	Zorunlu
KEY	merchant_key	string	Zorunlu
KEY	plan_code	string	Zorunlu
KEY	recurring_number	integer	Zorunlu

### Yetkilendirme

Yetkilendirme, bağlantı girişimine izin verildiğine dair doğrulamayı tanımlayan bir başlık anahtarıdır. Yöntem "Bearer" olmalıdır. Örnek değer:

Bearer

eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFjZjdhZDhkODYzYTgyMzQ4Nzk5NTFkYzFIMDZkZT  
UxYjU0NWRjYmU3MzRjMmQ1OGNkMWFOWE4YjliZTkzMjdIZGVmZDdlMDliIn0.eyJhdWQiOiIxNSIsImp0aSI6Im

**Accept:** İstemci tarafında ne tür bir gösterim istendiğini belirler. Değer "application/json" olmalıdır.

**merchant\_key:** Halkode tarafından sağlanan satıcının benzersiz anahtarıdır.

**plan\_code:** İlk ödeme sırasında oluşturulan benzersiz bir anahtardır (Bu durumda, plan\_code web kancası yanıtından alınmalıdır).

**recurring\_number:** Halkode tarafından her yinelemede gönderilen bir webHook isteğinden alınmalıdır.



Örnek Cevap:

#### Response body

```
{
  "status_code": 100,
  "message": "Recurring created successfully",
  "recurring_id": 31,
  "plan_code": "15978357430kZFft",
  "currency": "TRY",
  "currency_symbol": "₺",
  "first_amount": 20,
  "recurring_amount": 20,
  "total_amount": 120,
  "payment_number": 6,
  "payment_interval": 1,
  "payment_cycle": "DAY",
  "first_order_id": "15978356611194",
  "merchant_id": 56307,
  "card_no": "5406****5403",
  "next_action_date": "2020-09-04",
  "recurring_status": "Active",
  "transaction_date": "2020-08-19 14:15:45",
  "transactionHistories": [
    {
      "id": 99,
      "sale_recurring_id": 31,
      "sale_id": 16628,
      "merchant_id": 56307,
      "action_date": "2020-08-19 14:15:45",
      "status": "Success",
      "recurring_number": 1
    }
  ]
}
```

**Adım 3:** Sitenizde "recurring\_number" ve "status" değerlerine göre işlemler oluşturun.

### 7.1. Update Recurring

Yinelenen ödeme bu API kullanılarak güncellenebilir.

Metot	URL	İçerik Türü
POST	<ACCESS_URL>/api/recurringPlan/update	application/json

Tür	Parametre	Veri Türü	Koşul
HEADER	Authorization	string	Zorunlu
HEADER	Accept	string	Zorunlu
KEY	merchant_key	string	Zorunlu
KEY	plan_code	string	Zorunlu
KEY	recurring_payment_number	integer	Zorunlu
KEY	recurring_status	string	Opsiyonel





## 7.2. Query Recurring Information

Belirli bir yinelenen ödeme bilgisi bu API tarafından getirilebilir.

Metot	URL	İçerik Türü
POST	<ACCESS_URL>/api/recurringPlan/query	application/json

Tür	Parametre	Veri Türü	Koşul
HEADER	Authorization	string	Zorunlu
HEADER	Accept	string	Zorunlu
KEY	merchant_key	string	Zorunlu
KEY	plan_code	string	Zorunlu

## Yetkilendirme

Yetkilendirme, bağlantı girişimine izin verildiğine dair doğrulamayı tanımlayan bir başlık anahtarıdır. Yöntem "Bearer" olmalıdır. Örnek değer:

Bearer

eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVlZGFjZjhkODYzYTgyMzQ4Nzk5NTFkYzFIMDZkZT  
UxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkzMjdGVmZDdlMDIiIn0.eyJhdWQiOiIxNSIsImp0aSI6ImRlMGVlZGFjZjhkODYzYTgyMzQ4Nzk5NTFkYzFIMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4Yjl  
iZTkzMjdGVmZDdlMDIiIiwiaWF0IjoxNTczNzUyNDcyLCJuYmYiOiJlNzMyNTI0NzIsImV4cCI6MTYwNTM3NDgz  
Miwic3ViljoieMSIsInNjb3BlcyI6W119.

**Accept:** İstemci tarafında ne tür bir gösterim istendiğini belirler. Değer "application/json" olmalıdır.

**merchant\_key:** Halkode tarafından sağlanan satıcının benzersiz anahtarıdır.

**plan\_code:** İlk ödeme sırasında oluşturulan benzersiz bir anahtardır (Bu durumda, plan\_code web kancası yanıtından alınmalıdır).

### Örnek Request:

```
{
  "merchant_key": "$2y$10$w/ODdbTmfubcbUCUq/ia3OoJFMUmkM1UVNBilQIUlFUIPmaLUT1he",
  "plan_code": "1649229007oFOkJ8",
}
```

**Başarısız Response:**

```
{
  "status_code": 70,
  "message": "Recurring Not Found",
  "plan_code": "1649229007oFOkJ8df"
}
```



### Başarılı Response:

```
{
  "status_code": 100,
  "message": "Recurring card Found",
  "recurring_id": 328,
  "plan_code": "1649229007oFOkJ8",
  "currency": "TRY",
  "first_amount": 103,
  "recurring_amount": 100,
  "total_amount": 503,
  "payment_number": 5,
  "payment_interval": 1,
  "payment_cycle": "MONTH",
  "merchant_id": 98950,
  "recurring_status": "Active",
  "transaction_date": "2022-04-06T07:10:07.000000Z",
  "recurringCardList": [
    {
      "sale_recurring_id": 328,
      "card_token": "f02102a7-6d5e-4200-b259-5fcbcad06595",
      "card_user_key": "195f8bfb-9632-4ef5-aca6-2a9e441c878c",
      "status": "Active",
      "created_at": "2022-04-06T07:10:07.000000Z",
      "updated_at": "2022-04-06T07:10:07.000000Z"
    }
  ]
}
```

### 7.3. Delete Recurring

Bu API kullanılarak yinelenen bir ödeme silinebilir.

Metot	URL	İçerik Türü
POST	<ACCESS_URL>/api/recurringPlan/delete	application/json

Tür	Parametre	Veri Türü	Koşul
HEADER	Authorization	string	Zorunlu
HEADER	Accept	string	Zorunlu
KEY	merchant_key	string	Zorunlu
KEY	plan_code	string	Zorunlu



### Yetkilendirme

Yetkilendirme, bağlantı girişimine izin verildiğine dair doğrulamayı tanımlayan bir başlık anahtarıdır. Yöntem "Bearer" olmalıdır

Örnek Değer:

Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVIZGFjZjdhZDhkODYzYTgyMzQ4Nzk5NTFk

**Accept:** İstemci tarafında ne tür bir gösterim istendiğini belirler. Değer "application/json" olmalıdır

**merchant\_key:** Halkode tarafından sağlanan satıcının benzersiz anahtarıdır.

**plan\_code:** İlk ödeme sırasında oluşturulan benzersiz bir anahtardır (Bu durumda, plan\_code web kancası yanıtından alınmalıdır).

Örnek İstek :

```
{
  "merchant_key": "$2y$10$w/ODdbTmfubcbUCUq/ia3OoJFMUmkM1UVNBilQluLfUIPmaLUT1he",
  "plan_code": "1649229007oFOkJ8",
}
```

### Başarısız Response:

```
{
  "status_code": 72,
  "message": "Failed to delete"
}
```

### Başarılı Response:

```
{
  "status_code": 100,
  "message": "Recurring plan Deleted successfully"
}
```

### 7.4. Sale WebHook

Öncelikle, <http://app.halkode.com.tr/merchant/apisetting> adresindeki Halkode satıcı panelinde satış web kancası URL'nizi (anahtar, değer) ayarlamanız gerekir. Bu özelliği elde etmek için, satıcıların satın alma talebi gönderirken sale\_web\_hook\_keywith invoice göndermeleri gerekir. Bu anahtar isteğe bağlıdır, ancak gönderilirse geçerli bir anahtar olmalıdır.

Her ödeme için, satış web kancası url'nize aşağıda verilen parametrelerle bir POST isteği gönderiyoruz.

**Headers:**

'Content-Type: application/x-www-form-urlencoded'

Tip	Parametere	Örnek Değer
Key	invoice_id	8iu75g
Key	order_id	15767887576675
Key	status	“Completed” / “Failed”
KEY	payment_status	1=Completed, 0 = Failed
KEY	status_description	Transaction Failed due to .....
KEY	payment_method	1= Card, 2=Moblie, 3=Wallet
KEY	Hash_key	Gfh:rtete:tertetetrttyfghhgjjjggtgh=
KEY	transaction_type	“Auth” / “Pre-Authorization”

**Parametre Açıklaması**

Key	Explanation
payment_status	1/0 olabilir. 1=başarı, 0=başarısız
order_no	Halkode sipariş no
invoice_id	Üye işyeri fatura no
status_code	Halkode Durum kodu, 100 başarı kodudur
status_description	İşlem açıklaması
payment_method	1= Kredi Kartı, 2= Mobil, 3= Cüzdan
transaction_type	transaction_type == “Auth” //transaction amount is deducted from the card instantly.  transaction_type == “Pre-Authorization” transaction amount will be deducted from the card later.
error_code	Hata kodu
error	Hata açıklaması
hash_key	Talebin Halkode'dan geldiğini doğrulamak için kod
merchant_commission	Satış işleminden elde edilen satıcı komisyon
user_commission	Satış işleminden gelen kullanıcı bildirimi
md_status	Satış işleminden md_durumu





## 7.6. Pay With White Label Smart 3D

PaySmart3D API, sipariş ve kredi kartı bilgilerini Halkode ödeme entegrasyon sistemine göndermek için kullanılır. Satıcı web sitesi bir ödeme formu gönderdikten sonra kullanıcı banka sayfasına yönlendirilecektir. Ödeme, banka ağ geçidinde bir SMS kodu ile doğrulanacaktır. Ödeme başarılı olduğunda, kullanıcı satıcı başarı URL'sine geri yönlendirilecek, aksi takdirde satıcı tarafından belirlenen URL'yi iptal etmek için yönlendirilecektir. Bu ödeme API'sinde, diğer ödeme api çağrıları gibi getPos Api'yi çağdırmaya gerek yoktur.

**Not:** PaySmart3D Api'sini çağdırmak için ajax istekleri kullanmayın. "<ACCESS\_URL>/api/paySmart3D" ye normal form gönderimi olmalıdır.

### Request

Metot	URL	İçerik Türü
POST	<ACCESS_URL>/api/paySmart3D	application/x-www-form-urlencoded

Tür	Parametre	Veri Türü	Koşul
KEY	cc_holder_name	string   max:100	Zorunlu
KEY	cc_no	string	Zorunlu
KEY	expiry_month	string	Zorunlu
KEY	expiry_year	string	Zorunlu
KEY	cvv	string	Zorunlu
KEY	currency_id	string	Zorunlu
KEY	currency_code	string	Zorunlu
KEY	installments_number	number	Zorunlu
KEY	invoice_id	string	Zorunlu
KEY	invoice_description	string	Zorunlu
KEY	name	string	Zorunlu
KEY	surname	string	Zorunlu
KEY	total	double	Zorunlu
KEY	merchant_key	string	Zorunlu
KEY	items	string	Zorunlu
KEY	cancel_url	string	Zorunlu
KEY	return_url	string	Zorunlu
KEY	hash_key	string	Zorunlu
KEY	bill_address1	string	Opsiyonel
KEY	bill_address2	string	Opsiyonel
KEY	bill_city	string	Opsiyonel
KEY	bill_postcode	string	Opsiyonel
KEY	bill_state	string	Opsiyonel
KEY	bill_country	string	Opsiyonel
KEY	bill_email	string	Opsiyonel
KEY	bill_phone	string	Opsiyonel
KEY	sale_web_hook_key	string	Opsiyonel
KEY	card_program	string	Opsiyonel
KEY	ip	string	Opsiyonel
KEY	transaction_type	string	Opsiyonel
KEY	payment_completed_by	string	Opsiyonel



## Request for Recurring

KEY	order_type	Integer	Zorunlu
KEY	recurring_payment_number	Integer	Zorunlu
KEY	recurring_payment_cycle	string	Zorunlu
KEY	recurring_payment_interval	integer	Zorunlu
KEY	recurring_web_hook_key	string	Zorunlu

## Tarım için Request

KEY	maturity_period	Integer	Zorunlu
KEY	payment_frequency	Integer	Zorunlu
KEY	response_method	String: POST/GET	Opsiyonel

**Not:** Cancel\_url, ödeme başarısız olduğunda yönlendirmek için kullanılır ve return\_url, ödeme başarılı olduğunda yönlendirmek için kullanılır.

Ödemenin başarısız veya başarılı olmasına bakılmaksızın, iptal ve başarı url'si ile sorgu dizesinde aşağıdaki anahtarlar kullanılabilir:

payment\_status, order\_id ve invoice\_id. Örneğin, success URL'si [https://<my-domain.com/>](https://<my-domain.com/>?payment_status=1&order_no=234234232&invoice_id=73434) ise, başarılı ödemeden sonra Halkode aşağıdaki adrese yönlendirecektir.

[https://<my-domain.com/?payment\\_status=1&order\\_no=234234232&invoice\\_id=73434](https://<my-domain.com/?payment_status=1&order_no=234234232&invoice_id=73434)

Sorgu dizesinde, payment\_status = 1 ise, sepeti temizlemek ve sipariş durumunu "Tamamlandı" olarak değiştirmek için getOrderStatus API'si çağrılmalıdır.

**name:** Adı alanı. Örneğin, ürünü satın alan kişinin adı "John Dao" ise, ad "John" olmalıdır.

**surname:** Soyadı alanı. Örneğin, ürünü satın alan kişinin adı "John Dao" ise, soyadı "Dao" olmalıdır.

**sale\_web\_hook\_key:** İsteğe bağlı bir anahtardır. Bir satın alma isteği tamamlandığında, Halkode bir post isteği gönderir. Böylece satıcı kendi sitesinde bir etkinlik gerçekleştirebilir. Halkode, bu anahtarın veritabanında var olması gerektiğini doğrular. Satıcı, Halkode Satıcı Panelindeki Satış web kancası URL'sini bu anahtara göre atmalıdır.

**order\_type:** Eğer order\_type=1 ise, Halkode ödemeyi yineleme için doğrular. O zaman recurring\_payment\_number, recurring\_payment\_cycle, recurring\_payment\_intervalkeys boş olmamalıdır.

**card\_program:** "WORLD", "BONUS", "MAXIMUM", "BANKKART\_COMBO", "PARAF", "AXESS", "ADVANT", "CARD\_FNS" kart markalarından birisi olmalıdır.

**transaction\_type:** "PreAuth" için sistemde bekleyen bir işlem oluşturulur. Ancak daha sonra üye işyeri onayı ile Tamamlandı'ya dönüştürülür.

**payment\_completed\_by:** "Üye işyeri" veya "uygulama" olarak 2 değer alabilir.

"payment\_completed\_by" => "merchant" için, ödeme alma / yakalama / tamamlama aşaması satıcı tarafından CompletePayment API aracılığıyla gerçekleştirilmelidir. "app" durumunda (bu da değer olarak varsayılandır) ödeme PFMS/Application aracılığıyla tamamlanacaktır. Satıcı "merchant" gönderir ve ödemeyi API aracılığıyla tamamlamazsa, işlem 15 dakikalık bekleme süresinden sonra otomatik olarak başarısız ödemeye dönüştürülür.





**maturity\_period:** İşlemin tarım olması durumunda gönderilmesi gereken tam sayı. vade\_dönemi Değerler, tarım kartı veya bin numarası kullanılarak get installment api çağrısı ile elde edilebilir.

**payment\_frequency:** İşlemin tarım olması durumunda gönderilmesi gereken tam sayı. payment\_frequency Değerler, tarım kartı veya bin numarası kullanılarak get installment api çağrısı ile elde edilebilir.

**recurring\_payment\_number:** Taksit sayısını tanımlar. İlk\_tutar 100 \$ ve yinelenen\_ödeme\_sayısı 5 ise, toplam tutar 100 \$ \* 5 = 500 \$ olarak düşülecektir. (İşlem maliyeti her işlemle birlikte eklenebilir)

**recurring\_payment\_cycle:** Recurring\_payment\_interval parametresinin birim türünü tanımlar. Olası değerler: D /M/Y Örn: D: Günler, M: Aylar, Y: Yıllar

**recurring\_payment\_interval:** Aralık değerini tanımlar. Eğer recurring\_payment\_interval= 2 ve recurring\_payment\_cycle = "M" ise işlem her 2 ayda bir gerçekleşecektir.

**recurring\_web\_hook\_key:** Üye işyerlerinin yinelenen web kancası url'sini tanımlar. Halkode Üye İşyeri Panelinde bu anahtara karşılık bir URL atanmalıdır. Halkode, bu anahtarın veritabanında bulunması gerektiğini doğrular ve ödeme yinelandığında gerekli bir değerdir.

**hash\_key:** Ödemeyi güvence altına almak için beyan edilir. Son kullanıcı bankaya gitmeden önce ürün fiyatını değiştirebilir. Aşağıda hash anahtarını yazmak için kullanılan algoritma verilmiştir.

```
function generateHashKey($total,$installment,$currency_code,$merchant_key,$invoice_id,
$app_secret)
{
    $data = $total.'|'.$installment.'|'.$currency_code.'|'.$merchant_key.'|'.$invoice_id;
    $iv = substr(sha1(mt_rand()), 0, 16);
    $password = sha1($app_secret);
    $salt = substr(sha1(mt_rand()), 0, 4);
    $saltWithPassword = hash('sha256', $password . $salt);
    $encrypted = openssl_encrypt("$data", 'aes-256-cbc', "$saltWithPassword", null, $iv);
    $msg_encrypted_bundle = "$iv:$salt:$encrypted";
    $msg_encrypted_bundle = str_replace('/', '__', $msg_encrypted_bundle);
    return $msg_encrypted_bundle;
}
```

**response\_method:** Bu parametrenin değerine (GET/POST) bağlı olarak yanıt http GET veya POST olarak gönderilecektir. Varsayılan olarak GET'tir.

**Response Redirect:** Ödeme işleminden sonra, son kullanıcı aşağıda verilen parametreler kullanılarak satıcı web sitesine yönlendirilir. Yanıt doğrulaması için bir hash\_key parametresi gönderiyoruz. Satıcı, istek parametresini 11.5'te açıklanan hash\_key şifresi çözülmüş değerle karşılaştırabilir.

```
$params = [
    'payment_status' => $status,
    'order_no' => $order_id,
    'invoice_id' => invoice_id,
    'status_code' => $error_code,
    'status_description' => $status_message,
    'payment_method' => $payment_type,
    'transaction_type' => $transaction_type,
    'error_code' => $error_code,
    'error' => $status_message,
    'hash_key' => $hash_key,
];
```



**payment\_status:** 0 veya 1 olabilir. 0=Başarısız, 1=Tamamlandı

**order\_id:** Halkode sipariş id' si.

**invoice\_id:** Üye işyeri fatura id'sidir.

#### Response Parametre Açıklamaları

Anahtar	Açıklama
payment_status	1/0 olabilir. 1=başarı, 0=başarısız
order_no	Halkode sipariş no
invoice_id	Üye işyeri fatura id'si
status_code	Halkode Durum kodu, 100 başarı kodudur
status_description	İşlem açıklaması
payment_method	1= Kredi Kartı, 2= Mobil, 3= Cüzdan
transaction_type	transaction_type == "Auth" //işlem tutarı karttan anında düşülür. transaction_type == "Pre-Authorization" //işlem tutarı karttan daha sonra düşülecektir.
error_code	Hata kodu parametresinin değeri
error	Hata açıklaması
md_status	Kart kimlik doğrulama durumu
hash_key	Talebin Halkode'dan geldiğini doğrulamak için anahtar

#### Validate Response:

payment\_status == 1 and transaction\_type == "Auth" ///İşlem başarılıdır ve işlem tutarı karttan anında düşülür.

payment\_status == 1 and transaction\_type == "Pre-Authorization" //İşlem başarılıdır ve işlem tutarı karttan daha sonra çekilecektir.

payment\_status == 0 //İşlem başarısız

hash\_key, talebin Halkode'dan geldiğini doğrulamak için satıcı tarafında doğrulanmalıdır. Ayrıca Sipariş Durumu API'si (2.1) işlemin başarılı veya Başarısız olduğunu onaylar





```

    }
  ],
}

```

**Not:** Bir taksit altında bir kart programı için komisyon "x" gelirse, kart programı taksit için etkinleştirilmemiş demektir.

## 7.8. Hash Anahtarı Kullanarak Response Validation

3d ödemede, ödeme tamamlandıktan sonra, satıcı başarı veya başarısızlık bağlantısına yönlendirilir. Bu bağlantılarla ilgili sorun, anonim bir kişi tarafından erişilebilmesidir. Bu sorunu önlemek için Halkode'dan yönlendirme yapılırken linklere status, invoice\_id, order\_id ve hash\_key gibi bazı parametreler eklendiği için hash key kullanılarak isteğin doğrulanması önerilir.

Ayrıca, yinelenen ödemede, her yineleme için satıcı webhook'una bir post isteği gönderilir. Her zamanki gibi, bir webhook açık bir bağlantıdır. Dolayısıyla, yineleme istekleri hash\_key kullanılarak da doğrulanabilir. Doğrulama işlemi örnek kodu aşağıda verilmiştir.

```

function validateHashKey($hash_key, $secret_key){

    $status = $currency_code = '';
    $total = $invoice_id = $order_id = 0;

    if (!empty($hash_key)) {
        $hash_key = str_replace('_', '/', $hash_key);
        $password = sha1($secret_key);

        $components = explode(':', $hash_key);
        if (count($components) > 2) {
            $iv = $components[0] ?? '';
            $salt = $components[1] ?? '';
            $salt = hash('sha256', $password . $salt);
            $encrypted_msg = $components[2] ?? '';

            $decrypted_msg = openssl_decrypt($encrypted_msg, 'aes-256-cbc', $salt, null, $iv);

            if (strpos($decrypted_msg, '|') !== false){
                $array = explode('|', $decrypted_msg);
                $status = $array[0] ?? 0;
                $total = $array[1] ?? 0;
                $invoice_id = $array[2] ?? '0';
                $order_id = $array[3] ?? 0;
                $currency_code = $array[4] ?? '';
            }
        }
    }

    return [$status, $total, $invoice_id, $order_id, $currency_code];
}

```

Burada, \$hash\_key bir istekten alınmalıdır ve \$secret\_key Halkode'dan sağlanan bir satıcı uygulama sırrıdır.

**Not:** 3d ödeme için \$status = 0 (Başarısız) veya \$status = 1 (Tamamlandı). Ancak yinelenen web kancası ve satış web kancası için \$status = Tamamlandı veya \$status = Başarısız



### 7.9. Pay With White Label Smart 2D

Ödeme API'si, sipariş ve kredi kartı bilgilerini Halkode ödeme entegrasyon sistemine göndermek için kullanılır. Satıcı web sitesi ödeme sayfasını yüklemeyen ödeme durumunu hemen almalıdır. API başarı durumuna göre, sepet ve sipariş durumu buna göre değiştirilmelidir. Bu ödeme API'sinde, diğer ödeme api çağrıları gibi getPos Api'yi çağırmaya gerek yoktur.

#### Request

Metot	URL	İçerik Türü
<b>POST</b>	<ACCESS_URL>/api/paySmart2D	<i>application/json</i>

Tür	Parametre	Veri Türü	Koşul
KEY	cc_holder_name	string max:100	Zorunlu
KEY	cc_no	string	Zorunlu
KEY	expiry_month	string	Zorunlu
KEY	expiry_year	string	Zorunlu
KEY	cvv	string	Zorunlu
KEY	currency_id	string	Zorunlu
KEY	currency_code	string	Zorunlu
KEY	installments_number	number	Zorunlu
KEY	invoice_id	string	Zorunlu
KEY	invoice_description	string	Zorunlu
KEY	name	string	Zorunlu
KEY	surname	string	Zorunlu
KEY	total	double	Zorunlu
KEY	merchant_key	string	Zorunlu
KEY	items	string	Zorunlu
KEY	cancel_url	string	Opsiyonel
KEY	return_url	string	Opsiyonel
KEY	hash_key	string	Zorunlu
KEY	bill_address1	string	Opsiyonel
KEY	bill_address2	string	Opsiyonel
KEY	bill_city	string	Opsiyonel
KEY	bill_postcode	string	Opsiyonel
KEY	bill_state	string	Opsiyonel
KEY	bill_country	string	Opsiyonel
KEY	bill_email	string	Opsiyonel
KEY	bill_phone	string	Opsiyonel
KEY	sale_web_hook_key	string	Opsiyonel
KEY	card_program	string	Opsiyonel
KEY	ip	string	Opsiyonel
KEY	transaction_type	string	Opsiyonel
KEY	vpos_type	string	Opsiyonel
KEY	identity_number	string	Opsiyonel



## Request for Recurring

KEY	order_type	Integer	Zorunlu
KEY	recurring_payment_number	Integer	Zorunlu
KEY	recurring_payment_cycle	string	Zorunlu
KEY	recurring_payment_interval	integer	Zorunlu
KEY	recurring_web_hook_key	string	Zorunlu

**name:** Adı alanı. Örneğin, ürünü satın alan kişinin adı "John Dao" ise, ad "John" olmalıdır

**surname:** Soyadı alanı. Örneğin, ürünü satın alan kişinin adı "John Dao" ise, soyadı "Dao" olmalıdır.

**sale\_web\_hook\_key:** İsteğe bağlı bir anahtardır. Bir satın alma isteği tamamlandığında, HalkÖde bir gönderi isteği gönderir. Böylece satıcı kendi sitesinde bir etkinlik gerçekleştirebilir. Halkodevalidates bu anahtarın veritabanında var olması gerekir. Satıcı, HalkodeMerchant Panelindeki Satış web kancası URL'sini bu anahtara göre atmalıdır.

**order\_type:** Eğer bu alan 1 ise, Halk ödemeyi yineleme için doğrular. Bu durumda recurring\_payment\_number, recurring\_payment\_cycle, recurring\_payment\_interval anahtarları boş olmamalıdır.

**card\_program:** "WORLD", "BONUS", "MAXIMUM", "BANKKART\_COMBO", "PARAF", "AXESS", "ADVANT", "CARD\_FNS" markalarından biri olmalı.

**transaction\_type:** "PreAuth" için sistemde bekleyen bir işlem oluşturulur. Ancak daha sonra üye işyeri onayı ile Tamamlandı'ya dönüştürülür.

### Insurance payment:

**vpos\_type:** Yalnızca PaySmart2d çözümü için bir sigorta Satıcısı tarafından bir sigorta ödemesi başlatmak için geçerlidir. Sunulan değer "sigorta" olmalıdır, aksi takdirde bir doğrulama istisnası atacaktır  
**identity\_number :** sadece PaySmart2d Çözümü için "vpos\_type": "insurance" mevcut olduğunda bir sigorta Satıcısı tarafından bir sigorta ödemesi başlatmak için geçerlidir. Değer (tckn/vkn/tin) 10-11 hane arasında olmalıdır

**cc\_no:** Formatında gönderilmelidir: "first\_six\_digits + \*\*\*\*\* + last\_six\_digits", yani cc\_no 1234567897891234 ise cc\_no "123456\*\*\*\*\*1234" gibi gönderilmelidir (lütfen yıldız sayısını dikkate alın, her zaman altı (6) sayısı olarak gelmelidir), aksi takdirde istisna atılır. Veya, "first\_eight\_digits + \*\*\*\* + last\_four\_digits", yani cc\_no 1234567897891234 ise cc\_no "12345678\*\*\*\*1234" gibi gönderilmelidir (lütfen yıldız sayısını dikkate alın, her zaman dört (4) sayısı olarak gelmelidir), aksi takdirde istisna atılır.

cvv, expiry\_month, expiry\_year Sigorta ödemesi için isteğe bağlıdır.

**recurring\_payment\_number:** Taksit sayısını tanımlar. Eğer first\_amount 100\$ ise ve recurring\_payment\_number 5 ise, toplam tutar 100\$\*5=500\$ olarak düşülecektir. (İşlem maliyeti her işlemde eklenebilir)

**recurring\_payment\_cycle:** Recurring\_payment\_interval parametresinin birim türünü tanımlar.

Olası değerler: D /M/Y

Örn: D: Günler, M: Aylar, Y: Yıllar

**recurring\_payment\_interval:** Taksit aralık değerini tanımlar. Eğer recurring\_payment\_interval= 2 ve recurring\_payment\_cycle = "M" ise işlem her 2 ayda bir gerçekleşecektir.



**recurring\_web\_hook\_key:** Üye işyeri yinelenen web kancası url'sini tanımlar. Halkode Üye İşyeri Panelinde bu anahtara karşılık bir URL atanmalıdır. Halkode, bu anahtarın veritabanında bulunması gerektiğini doğrular ve ödeme yinelenildiğinde gerekli bir değerdir.

**hash\_key:** Ödemeyi güvence altına almak için beyan edilir. Son kullanıcı bankaya gitmeden önce ürün fiyatını değiştirebilir. Aşağıda hash anahtarını yazmak için kullanılan algoritma verilmiştir.

```
function generateHashKey($total,$installment,$currency_code,$merchant_key,$invoice_id,
$app_secret){
    $data = $total.'|'.$installment.'|'.$currency_code.'|'.$merchant_key.'|'.$invoice_id;
    $iv = substr(sha1(mt_rand()), 0, 16);
    $password = sha1($app_secret);
    $salt = substr(sha1(mt_rand()), 0, 4);
    $saltWithPassword = hash('sha256', $password . $salt);
    $encrypted = openssl_encrypt("$data", 'aes-256-cbc', "$saltWithPassword", null, $iv);
    $msg_encrypted_bundle = "$iv:$salt:$encrypted";
    $msg_encrypted_bundle = str_replace('/', '_', $msg_encrypted_bundle);
    return $msg_encrypted_bundle;
}
```

**Başarısız Response:**

```
{
    "status_code": 41,
    "status_description": "transaction failed",
    "data": {
        "order_no": "162616264070046",
        "invoice_id": "5edfg345ffgdfdgff",
        "payment_method": 1,
        "credit_card_no": "5355765990527226",
        "transaction_type": "Auth",
        "payment_status": 0,
        "payment_method": 1,
        "error_code": "",
        "error": "transaction failed",
        "hash_key":
        "808b7dd19e13efb6:2374:aoDf__tgqgM5fTuFJKFrEcxC4vTVz2uNarIGX31rzVTP+JZPTteiy5DGam__5wBVk"
    }
}
```

**Başarılı Response:**

```
{
  "status_code": 100,
  "status_description": "Payment process successful",
  "data": {
    "order_no": "162616268649431",
    "invoice_id": "5edfg345ffgdfgff",
    "payment_method": 1,
    "credit_card_no": "5355765990527226",
    "transaction_type": "Auth",
    "payment_status": 1,
    "payment_method": 1,
    "error_code": 100,
    "error": "Transaction Successful",
    "hash_key":
"aabb5f5b9f179594:7ade:JiXxJ7wo99Y__9Gm__Kg0lwP273bXxNftgO++vp9ykxyl7n7vmjOKGvtgMklus8mOk",
    "merchant_commission": "2.18"
  },
  "user_commission": "0.96",
}
```

**Response Parametreleri**

Anahtar	Açıklama
payment_status	1/0 olabilir. 1=başarı, 0=başarısız
order_no	Halkode sipariş no
invoice_id	Üye işyeri fatura id'si
status_code	Halkode Durum kodu, 100 başarı kodudur
status_description	İşlem açıklaması
payment_method	1= Kredi Kartı, 2= Mobil, 3= Cüzdan
transaction_type	transaction_type == "Auth" //işlem tutarı karttan anında düşülür. transaction_type == "Pre-Authorization" //işlem tutarı karttan daha sonra düşülecektir.
error_code	Hata kodu parametresinin değeri
error	Hata açıklaması
hash_key	İsteğin Halkode'dan geldiğini doğrulamak için kod
merchant_commission	Bir satış işleminden kazanılan üye işyeri komisyonu
user_commission	Bir satış işleminden kazanılan kullanıcı komisyonu
settlement_date	Satıcının bir işlemin ödeme tarihi



**Cevap:**

**Koşul 1 :** status\_code== 100 and transaction\_type == "Auth" // işlem başarılıdır ve işlem tutarı karttan anında düşülür.

**Koşul 2 :** status\_code== 100 and transaction\_type == "Pre-Authorization" //işlem başarılıdır ve işlem tutarı karttan daha sonra düşülecektir.

**7.10. Pay With CardToken**

PayByCardToken API'si sipariş göndermek için kullanılır. Satıcı web sitesi ödeme formunu gönderdikten sonra kullanıcı banka sayfasına yönlendirilecektir. Ödeme, banka ağ geçidinde bir SMS kodu ile doğrulanacaktır. Ödeme başarılı olduğunda, kullanıcı satıcı başarı URL'sine geri yönlendirilecek, aksi takdirde satıcı tarafından belirlenen URL'yi iptal etmek için yönlendirilecektir. Bu ödeme API'sinde, diğer ödeme api çağrıları gibi getPos Api'yi çağırmaya gerek yoktur.

**Not:** payByCardToken'ı çağırmak için ajax isteği kullanmayın.

"<ACCESS\_URL>/api/payByCardToken" için normal form gönderimi olmalıdır.

**Örnek Request**

Metot	URL	İçerik Türü
POST	<ACCESS_URL>/api/payByCardToken	application/x-www-form-urlencoded

Tür	Parametre	Veri Türü	Koşul
key	card_token	string	Zorunlu
Key	customer_number	number	Zorunlu
Key	customer_email	string	Zorunlu
Key	customer_phone	string	Zorunlu
Key	customer_name	string	Zorunlu
Key	customer_phone	string	Zorunlu
key	currency_code	string	Zorunlu
key	installments_number	number	Zorunlu
key	invoice_id	string	Zorunlu
key	invoice_description	string	Zorunlu
key	total	double	Zorunlu
key	merchant_key	string	Zorunlu
key	items	string	Zorunlu
key	cancel_url	string	Zorunlu
key	return_url	string	Zorunlu
key	hash_key	string	Zorunlu
key	bill_address1	string	Opsiyonel
key	bill_address2	string	Opsiyonel
key	bill_city	string	Opsiyonel
key	bill_postcode	string	Opsiyonel
key	bill_state	string	Opsiyonel
key	bill_country	string	Opsiyonel
key	bill_email	string	Opsiyonel
key	bill_phone	string	Opsiyonel
key	sale_web_hook_key	string	Opsiyonel
key	ip	string	Opsiyonel



## Request for Recurring

KEY	order_type	Integer	Zorunlu
KEY	recurring_payment_number	Integer	Zorunlu
KEY	recurring_payment_cycle	string	Zorunlu
KEY	recurring_payment_interval	integer	Zorunlu
KEY	recurring_web_hook_key	string	Zorunlu
KEY	is_remote_card_token	boolean	Opsiyonel (Varsayılan: false)

**Not:** Ödeme başarısız olduğunda yönlendirme yapmak için, return\_url ise ödeme başarılı olduğunda yönlendirme yapmak için kullanılır.

Başarısız veya başarılı ödemeden bağımsız olarak, iptal ve başarı url'si ile sorgu dizesinde aşağıdaki anahtarlar kullanılabilir:

payment\_status, order\_id ve invoice\_id. Örneğin, başarı URL'si https://<my-domain.com/ ise, başarılı ödemeden sonra Halkode https://<my-domain.com/?payment\_status=1&order\_no=234234232&invoice\_id=73434 adresine yönlendirecektir.

Sorgu dizesinde, payment\_status = 1 ise, sepeti temizlemek ve sipariş durumunu "Tamamlandı" olarak değiştirmek için getOrderStatus API'si çağrılmalıdır.

**name:** Adı alanı. Örneğin, ürünü satın alan kişinin adı "John Dao" ise, ad "John" olmalıdır.

**Surname:** Soyadı alanı. Örneğin, ürünü satın alan kişinin adı "John Dao" ise, soyadı "Dao" olmalıdır.

**sale\_web\_hook\_key:** İsteğe bağlı bir anahtardır. Bir satın alma isteği tamamlandığında, Halkode bir post isteği gönderir. Böylece satıcı kendi sitesinde bir etkinlik gerçekleştirebilir. Halkode, bu anahtarın veritabanında var olması gerektiğini doğrular. Satıcı, Halkode Satıcı Panelindeki Satış web kancası URL'sini bu anahtara göre atamalıdır.

**order\_type:** Eğer 1 ise, Halkode ödemeyi yineme için doğrular. O zaman recurring\_payment\_number, recurring\_payment\_cycle, recurring\_payment\_intervalkeys boş olmamalıdır.

**Card\_program:** "WORLD", "BONUS", "MAXIMUM", "BANKKART\_COMBO", "PARAF", "AXESS", "ADVANT", "CARD\_FNS" değerlerinden biri olmalıdır.

**recurring\_payment\_number:** Taksit sayısını tanımlar. İlk\_tutar 100 \$ ve yinelenen\_ödeme\_sayısı 5 ise, toplam tutar 100 \$ \* 5 = 500 \$ olarak düşülecektir. (İşlem maliyeti her işlemle birlikte eklenebilir).

**recurring\_payment\_cycle:** Recurring\_payment\_interval parametresinin birim türünü tanımlar. Olası değerler: D /M/Y Örn: D: Günler, M: Aylar, Y: Yıllar

**recurring\_payment\_interval:** Aralık değerini tanımlar. Eğer recurring\_payment\_interval= 2 ve recurring\_payment\_cycle = "M" ise işlem her 2 ayda bir gerçekleşecektir.

**recurring\_web\_hook\_key:** Üye işyeri yinelenen web kancası url'sini tanımlar. Halkode Üye İşyeri Panelinde bu anahtara karşılık bir URL atanmalıdır. Halkode, bu anahtarın veritabanında bulunması gerektiğini doğrular ve ödeme yinlendiğinde gerekli bir değerdir.

**is\_remote\_card\_token:** Card\_token'ın uzak sistem tarafından üretilip üretilmediğini tanımlar. Örneğin, kart belirteci uzak sistem tarafından oluşturulmuşsa is\_remote\_card\_token değeri doğrudur, aksi takdirde yanlıştır.

**hash\_key:** Ödemeyi güvence altına almak için beyan edilir. Son kullanıcı bankaya gitmeden önce ürün fiyatını değiştirebilir. Aşağıda hash anahtarını yazmak için kullanılan algoritma verilmiştir.



```

function generateHashKey($total,$installment,$currency_code,$merchant_key,$invoice_id,
    $app_secret)
{
    $data = $total.'|'.$installment.'|'.$currency_code.'|'.$merchant_key.'|'.$invoice_id;
    $iv = substr(sha1(mt_rand()), 0, 16);
    $password = sha1($app_secret);
    $salt = substr(sha1(mt_rand()), 0, 4);
    $saltWithPassword = hash('sha256', $password . $salt);
    $encrypted = openssl_encrypt("$data", 'aes-256-cbc', "$saltWithPassword", null, $iv);
    $msg_encrypted_bundle = "$iv:$salt:$encrypted";
    $msg_encrypted_bundle = str_replace('/', '___', $msg_encrypted_bundle);
    return $msg_encrypted_bundle;
}

```

#### Cevap:

**Koşul 1:** Payment\_status == 1 and transaction\_type == "Auth" // İşlem başarılıdır ve işlem tutarı karttan anında düşülür.

**Koşul 2:** Payment\_status == 1 and transaction\_type == "Pre-Authorization" //İşlem başarılıdır ve işlem tutarı karttan daha sonra çekilecektir.

**Koşul 3:** Payment\_status == 0 //İşlem başarısız

#### 7.11. SaveCard Api

Kart kaydetme API'si, kart bilgilerini Halkode Sistemine kaydetmek ve payByCardToken API'sinde kullanılacak yanıtta bir token döndürmek için kullanılır.

##### Request

Metot	URL	İçerik Türü
POST	<ACCESS_URL>/api/saveCard	application/json

Tür	Parametre	Veri Türü	Koşul
Key	merchant_key	string	Zorunlu
Key	card_holder_name	string max:100	Zorunlu
Key	card_number	number	Zorunlu
Key	expiry_month	string	Zorunlu
Key	expiry_year	string	Zorunlu
Key	customer_number	number unique	Zorunlu
Key	hash_key	string	Zorunlu
Key	customer_name	string	Opsiyonel
Key	customer_phone	string	Opsiyonel



**Yetkilendirme:** Bağlantı girişimine izin verildiğine dair doğrulamayı tanımlayan bir başlık anahtarıdır. Yöntem "Bearer" olmalıdır. Örnek değer:

Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRIMGVIZGFjZjdhZDhkODY

**Accept:** İstemci tarafında ne tür bir gösterim istendiğini belirler. Değer "application/json" olmalıdır

**merchant\_key:** Halkode tarafından sağlanan satıcının benzersiz anahtarıdır.

**cardholder\_name:** Kart sahibinin adı

**card\_number:** Kart numarası

**expiry\_month:** Geçerlilik süresi ay

**expiry\_year:** Geçerlilik süresi yıl

**customer\_number:** Kartı kaydedeceğimiz müşteri için, bu numara sayısal ve benzersiz olmalıdır

**customer\_name:** Müşteri adı kartı kaydedeceğimiz müşteri için

**customer\_phone:** Müşteri telefon numarası

**hash\_key:** Api isteğini güvence altına almak için bildirilir. Son kullanıcı, kartı kaydetmek için Halkode api'sine gitmeden önce verileri değiştirebilir. Aşağıda hash anahtarını yazmak için kullanılan algoritma verilmiştir.

```
function generateHashKey($merchant_key, $customer_number, $card_number, $card_holder_name,
$customer_number, $expiry_month, $expiry_year, $app_secret)
{
    $data = $merchant_key.'|'. $customer_number
    '|'. $card_holder_name.'|'. $card_number.'|'. $expiry_month.'|'. $expiry_year;
    $iv = substr(sha1(mt_rand()), 0, 16);
    $password = sha1($app_secret);
    $salt = substr(sha1(mt_rand()), 0, 4);
    $saltWithPassword = hash('sha256', $password . $salt);
    $encrypted = openssl_encrypt("$data", 'aes-256-cbc', "$saltWithPassword", null, $iv);
    $msg_encrypted_bundle = "$iv:$salt:$encrypted";
    $msg_encrypted_bundle = str_replace('/', '_', $msg_encrypted_bundle);
    return $msg_encrypted_bundle;
}
```

#### **Başarılı Request:**

```
{
  "status_code": "86",
  "status_description": "The card token save failed"
}
```

#### **Başarısız Request:**

```
{
  "status_code": 100,
  "status_description": "The card token saved successfully",
  "card_token": "WNPDZDQNMVNRQO23IAHKDKXIWCGRWHEXCFNRDXXXK5CMXGM5A"
}
```



## 7.12. Save Cart Edit Api

### Request

Metot	URL	İçerik Türü
<b>POST</b>	<ACCESS_URL>/api/editCard	<i>application/json</i>

Tür	Parametre	Veri Türü	Koşul
Key	merchant_key	string	Zorunlu
Key	card_token	string	Zorunlu
Key	customer_number	string	Zorunlu
Key	expiry_month	string	Zorunlu
Key	expiry_year	string	Zorunlu
Key	hash_key	string	Zorunlu
Key	card_holder_name	string max:100	Opsiyonel

**Yetkilendirme:** Bağlantı girişimine izin verildiğine dair doğrulamayı tanımlayan bir başlık anahtarındır. Yöntem "Bearer" olmalıdır. Örnek değer;

Bearer

eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVIZGFjZjhkdHkODYzYTgyMzQ4Nzk5NTFkYzFIMDZkZ

**Accept:** İstemci tarafında ne tür bir gösterim istendiğini belirler. Değer "application/json" olmalıdır

**merchant\_key:** Halkode tarafından sağlanan satıcının benzersiz anahtarındır.

**card\_token:** Halkode sistemine kaydedilen

**expiry\_month:** Kart son kullanım tarihi ay olarak

**expiry\_year:** Kart son kullanım tarihi yıl olarak

**customer\_number:** Kartı kaydedeceğimiz müşteri numarası, bu numara sayısal ve benzersiz olmalıdır

**hash\_key:** Kart kaydetme api isteğini güvence altına almak için bildirilir. Son kullanıcı, kartı kaydetmek için Halkode api'sine gitmeden önce verileri değiştirebilir. Aşağıda hash anahtarını yazmak için kullanılan algoritma verilmiştir.

**Accept:** İstemci tarafında ne tür bir gösterim istendiğini belirler. Değer "application/json" olmalıdır

**merchant\_key:** Halkode tarafından sağlanan satıcının benzersiz anahtarındır.

**card\_token:** Halkode kart token

**expiry\_month:** Son kullanma ay

**expiry\_year:** Son kullanma yıl

**customer\_number:** Kartı kaydedeceğimiz müşteri için customer\_number, bu numara sayısal ve benzersiz olmalıdır.



**hash\_key:** Api isteğini güvence altına almak için bildirilir. Son kullanıcı, kartı kaydetmek için Halkode api'sine gitmeden önce verileri değiştirebilir. Aşağıda hash anahtarını yazmak için kullanılan algoritma verilmiştir.

```
function generateHashKey($merchant_key, $customer_number, $card_token, $app_secret){  
$data = $merchant_key.'|'.$customer_number.'|'.$card_token  
$iv = substr(sha1(mt_rand()), 0, 16);  
$password = sha1($app_secret);  
$salt = substr(sha1(mt_rand()), 0, 4);  
$saltWithPassword = hash('sha256', $password . $salt);  
$encrypted = openssl_encrypt("$data", 'aes-256-cbc', "$saltWithPassword", null, $iv);  
$msg_encrypted_bundle = "$iv:$salt:$encrypted";  
$msg_encrypted_bundle = str_replace('/', '_', $msg_encrypted_bundle);  
return $msg_encrypted_bundle;  
}
```

**Başarısız Response:**

```
{  
  "status_code": 88,  
  "status_description": "The card token deletion failed",  
}
```

**Başarılı Response:**

```
{  
  "status_code": 100,  
  "status_description": "The card token updated successfully",  
  "card_token": "WNPDZDQNMVNRQO23IAHKDKXIWCGRWHEXCFNRDXXK5CMXGM5A"  
}
```

### 7.13. SaveCard Delete Api

**Request**

Metot	URL	İçerik Türü
POST	<ACCESS_URL>/api/deleteCard	application/json

Tür	Parametre	Veri Türü	Koşul
Key	merchant_key	string	Zorunlu
Key	card_token	string	Zorunlu
Key	customer_number	string	Zorunlu
Key	hash_key	string	Zorunlu

**Yetkilendirme:** Bağlantı girişimine izin verildiğine dair doğrulamayı tanımlayan bir başlık anahtarıdır. Yöntem "Bearer" olmalıdır. Örnek değer:



Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRIMGVIZGFjZjdhZDhkODYzYTgyMzQ4N

**Accept:** İstemci tarafında ne tür bir gösterim istendiğini belirler. Değer "application/json" olmalıdır

**merchant\_key:** Halkode tarafından sağlanan satıcının benzersiz anahtarıdır.

**card\_token:** Halkode ksilanacak kart token

**customer\_number:** Kartı kaydedeceğimiz müşteri için müşteri numarası, bu numara sayısal ve benzersiz olmalıdır

**hash\_key:** Kart kaydetme api isteğini güvence altına almak için bildirilir. Son kullanıcılar kartı kaydetmek için Halkode api'sine gitmeden önce verileri değiştirebilir. Aşağıda hash anahtarını yazmak için kullanılan algoritma verilmiştir.

**function generateHashKey**(\$merchant\_key, \$customer\_number, \$card\_token, \$app\_secret)

```
{
    $data = $merchant_key.'|'.$customer_number.'|'.$card_token
    $iv = substr(sha1(mt_rand()), 0, 16);
    $password = sha1($app_secret);
    $salt = substr(sha1(mt_rand()), 0, 4);
    $saltWithPassword = hash('sha256', $password . $salt);
    $encrypted = openssl_encrypt("$data", 'aes-256-cbc', "$saltWithPassword", null, $iv);
    $msg_encrypted_bundle = "$iv:$salt:$encrypted";
    $msg_encrypted_bundle = str_replace('/', '___', $msg_encrypted_bundle);
    return $msg_encrypted_bundle;
}
```

**Başarısız Response:**

```
{
  "status_code": 41,
  "status_description": "Invalid merchant Key",
}
```

**Başarılı Response:**

```
{
  "status_code": 100,
  "status_description": "Card deleted successfully",
}
```



## 7.14. SaveCard : Get Card Tokens Api

### Request

Metot	URL	İçerik Türü
POST	<ACCESS_URL>/api/getCardTokens	application/json

Tür	Parametre	Veri Türü	Koşul
Key Key	merchant_key customer_number	string string	Zorunlu Zorunlu

**Yetkilendirme:** Bağlantı girişimine izin verildiğine dair doğrulamayı tanımlayan bir başlık anahtarudur. Yöntem "Bearer" olmalıdır

Örnek değer:

Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRIMGVIZGFjZjdhZDhkODYzYTgyMzQ4

**Accept:** İstemci tarafında ne tür bir gösterim istendiğini belirler. Değer "application/json" olmalıdır

**merchant\_key:** Halkode tarafından sağlanan satıcının benzersiz anahtarudur.

**customer\_number:** Kartı kaydedeceğimiz müşteri numarası, bu numara sayısal ve benzersiz olmalıdır.

### Failed Response:

```
{  
  "status_code": 41,  
  "status_description": "Invalid merchant Key",  
}
```





#### Success Response:

```
{
  "status_code": 100,
  "status_description": "Data fetched successfully",
  "data": [
    {
      "id": 11,
      "pos_id": 0,
      "card_token": "KELUUKPQBCGYLV7FP7I7VUQNZP7N5UX7JUCPKZVOPKEMCEQP",
      "merchant_id": 98950,
      "customer_number": "1122343443-98950",
      "customer_name": "Taygun Alban",
      "customer_email": "taigun@gmail.com",
      "customer_phone": "+8801749452019",
      "bin": "540668",
      "created_at": "2021-04-08T14:07:42.000000Z",
      "updated_at": "2021-04-08T14:07:42.000000Z"
    }
  ]
}
```

#### 7.15. Get Transactions

API, belirli bir tarihe göre satış işlemlerini almak için kullanılabilir.

#### Request

Metot	URL	İçerik Türü
POST	<ACCESS_URL>/api/getTransactions	application/json

Tür	Parametre	Veri Türü	Koşul
HEADER	Authorization	string	Zorunlu
HEADER	Accept	string	Zorunlu
HEADER	Content-Type	string	Zorunlu
KEY	merchant_key	string	Zorunlu
KEY	hash_key	string	Zorunlu
KEY	date	string	Opsiyonel
KEY	invoiceid	string	Opsiyonel
KEY	currency_id	int	Opsiyonel
KEY	paymentmethodid	int	Opsiyonel
KEY	minamount	int	Opsiyonel
KEY	maxamount	int	Opsiyonel
KEY	transactionState	int	Opsiyonel
KEY	type	string	Opsiyonel
KEY	order_by_option	int	Opsiyonel
KEY	from_date	datetime	Opsiyonel
KEY	to_date	datetime	Opsiyonel





**type:** String değer. Bu alanın değeri "günlük" veya "saatlik" veya "aralık" olabilir. İstek gövdesinde tür alanı mevcut değilse, sistem "günlük" olarak kabul edecektir.

**order\_by\_option:** Boş geçilebilir. Enum : aşağıda açıklandığı gibi

1 (varsayılan) => aralık sorgusu "from\_date" ve "to\_date" kullanarak oluşturma tarihine bakacaktır.

2 => aralık sorgusu "from\_date" ve "to\_date" kullanarak değişiklik tarihine bakacaktır.

**from\_date:** Bu bir datetime alanıdır. Type alanının değeri "range" ise from\_date alanı zorunludur. Aksi takdirde isteğe bağlıdır. Geçerli tarih-saat biçimi: 2021-04-20 21:00:00

Tarih aralığına göre filtreleme için kullanılabilir.

**to\_date:** Bu bir datetime alanıdır. Type alanının değeri "range" ise from\_date alanı zorunludur. Aksi takdirde isteğe bağlıdır. Geçerli tarih-saat biçimi: 2021-04-20 21:00:00

Tarih aralığına göre filtreleme için kullanılabilir.

#### **Post Request Params:**

```
{
  "merchant_key": "$2y$10$w/ODdbTmfubcbUCUq/ia3OoJFMUmKM1UVNBilQluLfUIPmaLUT1he",
  "hash_key": "823e40a698c3ca1f:4c80:cYTC/9s5ehGRtGeYyXA0Tw==",
  "date": "2021-02-10",
  "invoiceid": "",
  "currency_id": "",
  "paymentmethodid": "",
  "minamount": "",
  "maxamount": "",
  "transactionState": "",
  "type": daily
}
```

#### **Başarısız Response:**

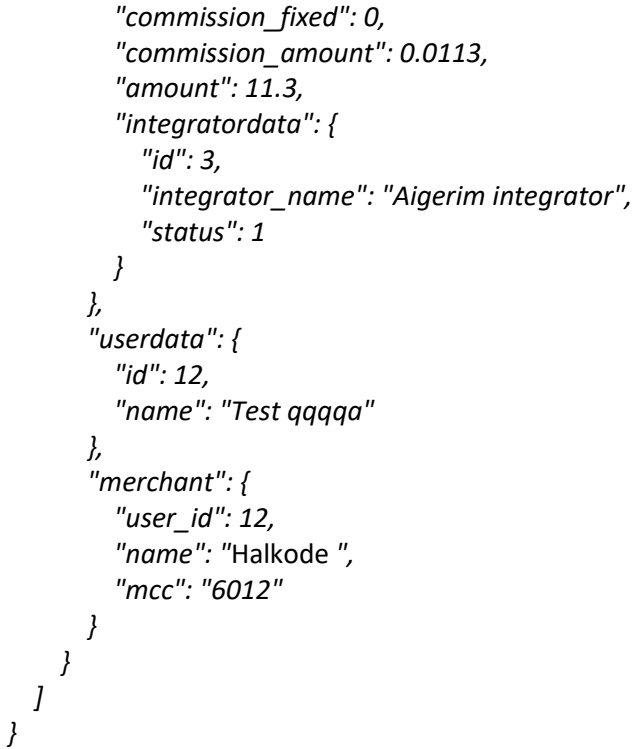
```
{
  "status_code": 90,
  "status_description": "Invalid Hash key"
}
```

#### **Başarılı Response:**

```
{
  "status_code": 100,
  "status_description": "Success",
  "data": [
    {
      "id": 275018,
      "user_id": 12,
      "merchant_id": 98950,
      "payment_id": "W7iQ8-M7jb-TC10-98950-100221",
      "transaction_state_id": 11,
      "merchant_commission": 0.665,
      "gross": 11.3,
      "net": 10.635,
      "payment_type_id": 1,
      "product_price": 10,
      "total_refunded_amount": 10,
      "card_holder_bank": "T. VAKIFLAR BANKASI T.A.O.",
    }
  ]
}
```



```
"operator": "",
"user_name": "",
"dpl_id": 0,
"currency_id": 1,
"invoice_id": "3905388578454743445487",
"credit_card_no": "535576****7226",
"installment": 1,
"rolling_amount": 1.13,
"gsm_number": "",
"payment_source": 12,
"card_program": null,
"result": "Approved(21041MekI07029311)",
"settlement_date_merchant": "2021-02-11 14:00:00",
"created_at": "2021-02-10T12:30:37.000000Z",
"updated_at": "2021-03-09T15:14:27.000000Z",
"recurring_id": 0,
"order_id": "16129494361217",
"ip": "103.195.141.79",
"auth_code": "175705",
"card_issuer_name": null,
"refunded_chargeback_fee": 50,
"pos_name": "ISBANK - Günlük TRY MAXIMUM",
"settlement_date_bank": "2021-02-11 14:00:00",
"transaction_state_label": "Chargebacked",
"payment_type_label": "MP Credit Card",
"currency_code": "TRY",
"rolling_balance": {
  "id": 10847,
  "sale_id": 275018,
  "effective_date": "2021-08-10 11:00:00"
},
"sale_billing": {
  "id": 15043,
  "sale_id": 275018,
  "card_holder_name": "A*****"
},
"merchant_sale": {
  "id": 161297,
  "sale_id": 275018,
  "merchant_commission_percentage": 5,
  "merchant_commission_fixed": 0.1,
  "end_user_commission_percentage": 3,
  "end_user_commission_fixed": 1,
  "merchant_rolling_percentage": 10,
  "cot_percentage": 0.75,
  "cot_fixed": 0
},
"refund_history": [],
"sale_recurring": null,
"sale_recurring_history": [],
"sale_integrator": {
  "id": 2100,
  "sale_id": 275018,
  "integrator_id": 3,
  "commission_percentage": 0.1,
```



PreAuth Payment'da sistemde Pending olarak bir işlem oluşturulur. Bu işlemi Failed veya Completed yapmak için aşağıda verilen istek parametreleri kullanılarak bu API çağrılmalıdır.

Metot	URL	İçerik Türü
POST	<ACCESS_URL>/api/confirmPayment	application/json

eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImRlMGVLZGFjZjhkODYzYTgyMzQ4Nzk5NTFkYzYlMDZkZT  
UxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4YjliZTkzMjdGVmZDdlMDIlIn0.eyJhdWQiOiILXSlSimp0aSI6Im  
RlMGVLZGFjZjhkODYzYTgyMzQ4Nzk5NTFkYzYlMDZkZTUxYjU0NWRjYmU3MzRjMmQ1OGNkMWFlOWE4Yjl  
iZTkzMjdGVmZDdlMDIiwiaWF0IjoxNTczNzUyNDcyLCJuYmYiOiJlNzNmM3NTI0NzlsImV4cCI6MTYwNTM3NDg3  
Miwic3ViljoiiMSIsbnNjb3BlcyI6W119.mDtdzcV15p8SnYjZYJUrhdsKO5NohXbkCAxKWWZ72INtrg86RZ1yxQwfQIR  
u6IPoa1rfG3M4ifsNeH



**Accept:** İstemci tarafında ne tür bir gösterim istendiğini belirler. Değer "application/json" olmalıdır

**invoice\_id:** Satıcı tarafından gönderilen benzersiz bir sipariş kimliğidir

**merchant\_key:** Halkode tarafından sağlanan satıcının benzersiz anahtarıdır.

**Status:** 1 veya 2 olmalı (1=onay, 2 = iptal)

**hash\_key:** Girdi değeri kullanılarak oluşturulması gereken benzersiz bir anahtardır. Aşağıda hash\_key oluşturma yöntemi verilmiştir.

```
function generateConfrimPaymentHashKey($merchant_key, $invoice_id, $status, $app_secret)
{
    $data = $merchant_key . '|' . $invoice_id . '|' . $status;
    $iv = substr(sha1(mt_rand()), 0, 16);
    $password = sha1($app_secret);
    $salt = substr(sha1(mt_rand()), 0, 4);
    $saltWithPassword = hash('sha256', $password . $salt);
    $encrypted = openssl_encrypt(
        "$data", 'aes-256-cbc', "$saltWithPassword", null, $iv );
    $msg_encrypted_bundle = "$iv:$salt:$encrypted";
    $msg_encrypted_bundle = str_replace('/', '___', $msg_encrypted_bundle);
    return $msg_encrypted_bundle;
}
```

**total:** Parametresinin değeri, ön yetkilendirme işleminin ürün fiyatına eşit veya daha az olmalıdır

#### **Başarısız Response:**

```
{
    "status_code": 105,
    "status_description": "The transaction is not Approved",
    "transaction_status": "Pending",
    "order_id": "162435924998223",
    "invoice_id": "50781624359247"
}
```

#### **Başarılı Response:**

```
{
    "status_code": 100,
    "status_description": "An order has been taken place for this invoice id: 33491162435928",
    "transaction_status": "Completed",
    "order_id": "162435932934307",
    "invoice_id": "33491162435928"
}
```

### **7.17. Pay With Card Token (Non-Secure)**

payByCardToken (güvenli olmayan) API, Halkode ödeme entegrasyon sistemine sipariş göndermek için kullanılır. Satıcı web siteleri, ödeme sayfasını yüklemeyen ödeme durumunu hemen almalıdır. API başarı durumuna bağlı olarak, sepet ve sipariş durumu buna göre değiştirilmelidir. Bu ödeme API'sinde, diğer ödeme API çağrıları gibi getPos Api'yi çağırmaya gerek yoktur.



### Request

Metot	URL	İçerik Türü
POST	<ACCESS_URL>/api/payByCardTokenNonSecure	application/json

Tür	Parametre	Veri Türü	Koşul
key	card_token	string	Zorunlu
Key	customer_number	number	Zorunlu
Key	customer_email	string	Zorunlu
Key	customer_phone	string	Zorunlu
Key	customer_name	string	Zorunlu
Key	customer_phone	string	Zorunlu
key	currency_code	string	Zorunlu
key	installments_number	number	Zorunlu
key	invoice_id	string	Zorunlu
key	invoice_description	string	Zorunlu
key	total	double	Zorunlu
key	merchant_key	string	Zorunlu
key	items	string	Zorunlu
key	hash_key	string	Zorunlu
key	bill_address1	string	Opsiyonel
key	bill_address2	string	Opsiyonel
key	bill_city	string	Opsiyonel
key	bill_postcode	string	Opsiyonel
key	bill_state	string	Opsiyonel
key	bill_country	string	Opsiyonel
key	bill_email	string	Opsiyonel
key	bill_phone	string	Opsiyonel
key	sale_web_hook_key	string	Opsiyonel
key	ip	string	Opsiyonel

### Request for Recurring

KEY	order_type	Integer	Zorunlu
KEY	recurring_payment_number	Integer	Zorunlu
KEY	recurring_payment_cycle	string	Zorunlu
KEY	recurring_payment_interval	integer	Zorunlu
KEY	recurring_web_hook_key	string	Zorunlu
KEY	is_remote_card_token	boolean	Opsiyonel (Default: false)

### Not:

**Name:** Adı alanı. Örneğin, ürünü satın alan kişinin adı "John Dao" ise, ad "John" olmalıdır

**Surname:** Soyadı alanı. Örneğin, ürünü satın alan kişinin adı "John Dao" ise, soyadı "Dao" olmalıdır.



### sale\_web\_hook\_key

*sale\_web\_hook\_key* is an Opsiyonel key. When a purchase request is completed, Halkode sends a post request. So that merchant can perform an event on their site. Halkode validates that this key must exist in the database. Merchant must assign the Sale web hook URL on the Halkode Merchant Panel against this key.

### order\_type

If *order\_type=1*, Halkode validates payment for recurring. Then *recurring\_payment\_number*, *recurring\_payment\_cycle*, *recurring\_payment\_interval* keys should not be empty.

### transaction\_type

For *transaction\_type* "PreAuth", a pending transaction is created in the system. But later it is converted to Completed by the merchant's confirmation.

### recurring\_payment\_number

*recurring\_payment\_number* defines installment count. If *first\_amount* is \$100 and *recurring\_payment\_number* is 5, then the total amount will be deducted as  $\$100 \times 5 = \$500$ . (Cost of transaction may be added with each transaction)

### recurring\_payment\_cycle

*recurring\_payment\_cycle* defines the unit type of *recurring\_payment\_interval* parameter. Possible values: D /M/Y

e.g: **D**: Days, **M**: Months, **Y**: Years

### recurring\_payment\_interval

*recurring\_payment\_interval* defines interval value. If *recurring\_payment\_interval= 2* and *recurring\_payment\_cycle = "M"* then transaction will occur once in every 2 months.

### recurring\_web\_hook\_key

*recurring\_web\_hook\_key* defines merchant recurring web hook url . An URL must be assigned on the Halkode Merchant Panel against this key. Halkode validates this key must exist in the database and it is a required value when payment is recurring.

### is\_remote\_card\_token

*is\_remote\_card\_token* defines provided *card\_token* is generated by remote system or not. For example, if the card token is generated by remote system then the value of *is\_remote\_card\_token* is true else it is false.

### hash\_key

*hash\_key* is declared to secure the payment. End user may change the product price before going to the bank. Here is the algorithm to write the hash key given below.

**function generateHashKey(\$total,\$installment,\$currency\_code,\$merchant\_key,\$invoice\_id,\$app\_secret)**

```
{
    $data = $total.'|'.$installment.'|'.$currency_code.'|'.$merchant_key.'|'.$invoice_id;
    $iv = substr(sha1(mt_rand()), 0, 16);
    $password = sha1($app_secret);
    $salt = substr(sha1(mt_rand()), 0, 4);
    $saltWithPassword = hash('sha256', $password . $salt);
    $encrypted = openssl_encrypt("$data", 'aes-256-cbc', "$saltWithPassword", null, $iv);
    $msg_encrypted_bundle = "$iv:$salt:$encrypted";
    $msg_encrypted_bundle = str_replace('/', '_', $msg_encrypted_bundle);
    return $msg_encrypted_bundle;
}
```



**Başarısız Response:**

```
{
  "status_code": 41,
  "status_description": "transaction failed",
  "data": {
    "order_no": "162616264070046",
    "invoice_id": "5edfg345fffgdgdgff",
    "payment_method": 1,
    "credit_card_no": "5355765990527226",
    "transaction_type": "Auth",
    "payment_status": 0,
    "payment_method": 1,
    "error_code": "",
    "error": "transaction failed",
    "hash_key":
      "808b7dd19e13efb6:2374:aoDf__tgqgM5fTuFJKFrEcxC4vTVz2uNarIGX31rzVTP+JZPTteiy5DGam__5wBVk"
  }
}
```

**Başarılı Response:**

```
{
  "status_code": 100,
  "status_description": "Payment process successful",
  "data": {
    "order_no": "162616268649431",
    "invoice_id": "5edfg345fffgdgdgff",
    "payment_method": 1,
    "credit_card_no": "5355765990527226",
    "transaction_type": "Auth",
    "payment_status": 1,
    "payment_method": 1,
    "error_code": 100,
    "error": "Transaction Successful",
    "hash_key":
      "aabb5f5b9f179594:7ade:JiXxJ7wo99Y__9Gm__Kg0lwP273bXxNftgO++vp9ykxyl7n7vmjOKGvtgMklus8mOk"
  }
}
```



### Response Parameter Explanation

Anahtar	Açıklama
payment_status	1/0 olabilir. 1=başarı, 0=başarısız
order_no	Halkode sipariş no
invoice_id	Üye işyeri fatura id
status_code	Halkode Durum kodu, 100 başarı kodudur.
status_description	İşlem açıklaması
payment_method	1= Kredi Kartı, 2= Mobil, 3= Cüzdan
transaction_type	transaction_type == "Auth" //işlem tutarı karttan anında düşülür. transaction_type == "Pre-Authorization" //işlem tutarı karttan daha sonra düşülecektir.
error_code	Hata kod parametresinin değeri
error	Hata açıklaması
hash_key	Talebin Halkode'dan geldiğini doğrulamak için

### Validate Response:

**Koşul 1:** status\_code== 100 and transaction\_type == "Auth" // işlem başarılıdır ve işlem tutarı karttan anında düşülür.

**Koşul 2:** status\_code== 100 and transaction\_type == "Pre-Authorization" //işlem başarılıdır ve işlem tutarı karttan daha sonra düşülecektir.

### 7.18. All Transaction API

API, sağlanan istek parametreleri ile satış işlemlerini almak için kullanılabilir

#### Request

Metot	URL	İçerik Türü
POST	<ACCESS_URL>/api/alltransaction	application/json





**order\_id:** Satış siparişi id Örnek '164290959020713

**payment\_method:** Ödeme yöntemi. 1= Kredi Kartı, 2= Mobil, 3= Cüzdan

**date\_range:** Tarih formatı örnek- '2021-04-20 - 2021-04-20' vb. gibi olmalıdır.

**currency\_code:** Varsayılan olarak "TRY" olarak ayarlanmıştır. Halkode Yalnızca "TRY", "USD", "EUR" değerlerine izin verir

**amount:** int degeri, boş olabilir. '10.00'

**minamount:** int degeri, boş olabilir.

**maxamount:** int degeri, boş olabilir.

**billing\_phone:** String degeri, boş olabilir.

**billing\_email:** String degeri, boş olabilir.

**transactionState:** int degeri, boş olabilir. 1= Tamamlandı, 3 = Beklemede

**search\_key:** int, string değer. Order\_id, invoice\_id, merchant\_name vb. gibi bazı sütun adlarına göre arama yapılır.

**order:** String değer, asc/desc

**page\_limit:** int, string değer.10,20...

Headers
Authorization: Bearer Content-Type: application/json
Request
<pre>{   "transaction_id": "",   "invoice_id": "",   "order_id": "",   "payment_method": "",   "transactionState": "",   "date_range": "2021/12/23 - 2022/01/23",   "currency_code": "TRY",   "amount": "",   "maxamount": "",   "minamount": "",   "billing_phone": "",   "billing_email": "",   "search_key": "vPiWL-pMmb-TC10-98950-230122",   "order": "desc",   "page_limit": "2" }</pre>



## Response

### Response

#### Successful

```
{
  "status_code": 100,
  "status_description": "data fetched successfully",
  "data": {
    "current_page": 1,
    "data": [
      {
        "id": 287447,
        "payment_id": "vPiWL-pMmb-TC10-98950-230122",
        "invoice_id": "14391",
        "order_id": "164290959020713",
        "remote_order_id": "",
        "user_id": 12,
        "end_user_id": 0,
        "user_name": "",
        "merchant_id": 98950,
        "merchant_name": "Fintlix Demo",
        "transaction_state_id": 3,
        "purchase_id": 0,
        "gross": 5,
        "refund_request_amount": 0,
        "fee": 0.1,
        "net": 4.9,
        "pay_by_token_fee": 0,
        "cost": 0.05,
        "rolling_amount": 0,
        "product_price": 5,
        "user_commission": 0,
        "merchant_commission": 0.1,
        "gsm_number": "",
        "settlement_date_bank": "",
        "settlement_date_merchant": "",
        "payment_type_id": 1,
        "operator": "",
        "issuer_name": "DENIZBANK A.S.",
        "card_program": "",
        "card_issuer_name": "",
        "pos_name": "Deniz DCC Try Pos",
        "pos_id": 113,
        "allocation_id": 0,
        "campaign_id": 0,
        "dpl_id": 0,
        "json_data": "{\"action\":\"WL_REQUEST_PAY_SMART_3D\",\"transaction_type\":\"Auth\",\"merchant_key\":\"$2y$10$w\\ODdbTmfubcbUCUq\\V\\ia3OoJFMUmK1UVNBilQIuLfUIPmaLUT1he\",\"invoice_id\":\"14391\",\"total\":\"5\",\"items\":{\"name\":\"Test Product\",\"price\":\"5.00\",\"quantity\":1,\"description\":\"Some Description\"}},\"currency_code\":\"TRY\",\"cc_no\":\"cj>_uWL@_fv<2ad9:tE@_:bXl37OOAv0mU4GU4PngAsnLel6plezOQR\\kZtJeryKE=\",\"installments_number\":1,\"cancel_url\":\"https://\\localhost\\app.halkode.com.tr\\api\\payment_success.php?merchant_key=%242y%2410%24w%2FODdbTmfubcbUCUq%2Fia3OoJFMUmK1UVNBilQIuLfUIPmaLUT1he\",\"return_url\":\"https://\\loc
```



```
alhost\\Vapp.halkode.com.tr\\Vapi\\Vpayment_success.php?merchant_key=%242y%2410%24w%2FODdbTmfubcbUCUq%2Fia3OoJFMUmK1UVNBilQluLfUIPmaLUT1he\\",\\"hash_key\\":\\"645e82f0848a9279:e319:x2odQXfr+xcml00SE6qMxs81SJiAu9p+nQjaY618glpH5lo6+4pFL68rNZYZkz+B4l4+l3JpF2zIEGG7k8pM63G0+jbsct+gEVv3cbjNEvg=\\",\\"name\\":\\"Customer Name\\",\\"surname\\":\\"Customer surname\\",\\"saved_card\\":0,\\"maturity_period\\":\\"0\\",\\"payment_frequency\\":\\"0\\",\\"pf_id\\":\\"\\",\\"bill_address1\\":\\"Address 1 should not more than 100\\",\\"bill_address2\\":\\"Address 2\\",\\"bill_city\\":\\"Istanbul\\",\\"bill_postcode\\":\\"1111\\",\\"bill_state\\":\\"Istanbul\\",\\"bill_country\\":\\"TURKEY\\",\\"bill_phone\\":\\"0088017771111\\",\\"bill_email\\":\\"demo@your-domain.com\\",\\"sale_web_hook_key\\":\\"\\",\\"merchant_server_id\\":\\"\\",\\"referer_url\\":\\"https:\\\\localhost\\Vapp.halkode.com.tr\\Vpublic\\Vapi\\VpaySmart3D\\Vpayment_page.php\\",\\"api_name\\":\\"PAY_SMART_3D\\",\\"payment_type\\":\\"3d\\",\\"is_pay_by_marketplace\\":0,\\"currency_id\\":1,\\"pos_id\\":113,\\"campaign_id\\":0,\\"allocation_id\\":0,\\"is_tarim_payment\\":false,\\"installment\\":1,\\"merchant_ip\\":\\"\\",\\"part_2\\":\\"cJ>_uWL@_fv<2ad9:tE@_:jocdjiBevAneTYND6J6mOg==\\",\\"part_1\\":\\"cJ>_uWL@_fv<2ad9:tE@_:oFAEwm\\VgdXomWrR\\VQofemQ=\\",
```

```
"document": "",
"total_refunded_amount": 0,
"refund_reason": "",
"refund_request_date": "",
"admin_process_date": "",
"chargeback_reject_explanation": "",
"created_at": "2022-01-23T03:46:30.000000Z",
"updated_at": "2022-01-23T03:46:30.000000Z",
"currency_id": 1,
"currency_symbol": "₺",
"installment": 1,
"maturity_period": 0,
"payment_frequency": 0,
"ip": "::1",
"result": "",
"credit_card_no": "546938****2434",
"card_holder_bank": "",
"sale_version": 0,
"sale_type": 1,
"payment_source": 5,
"refunded_chargeback_fee": 0,
"is_bank_refund_failed": 0,
"recurring_id": 0,
"sale_web_hook_key": "",
"auth_code": "",
"migration_status": 3,
"is_comission_from_user": 0,
"sale_recurring": "",
"paymenttype": {
  "id": 1,
  "name": "Credit Card",
  "created_at": "2019-07-31T02:52:28.000000Z",
  "updated_at": "2019-07-31T02:52:28.000000Z"
},
"rolling_balance": "",
"sale_integrator": "",
"sale_billing": "",
"sale_property": ""
```



```

    }
  ],
  "first_page_url": "https://localhost/app.halkode.com.tr/ccpayment/api/alltransaction?p
age=1",
  "from": 1,
  "last_page": 1,
  "last_page_url": "https://localhost/app.halkode.com.tr/ccpayment/api/alltransaction?p
age=1",
  "next_page_url": "",
  "path": "https://localhost/app.halkode.com.tr/ccpayment/api/alltransaction",
  "per_page": "2",
  "prev_page_url": "",
  "to": 1,
  "total": 1
}
}

```

#### Failure

```

{
  "status_code": 99
  "status_description": "failed",
  "data": []
}

```

### 7.19. Merchant 3D Model : CompletePayment Api

Bu, "payment\_completed\_by" : "merchant" ve "md\_status" : 1 ile bir işlem başlatıldıktan sonra satıcı tarafından ödeme alma adımının yapılabileceği ödeme yakalama apisidir.

#### Request

Metot	URL	İçerik Türü
POST	<ACCESS_URL>/payment/complete	application/json

Tür	Parametre	Veri Türü	Koşul
HEADER	Authorization	string	Zorunlu
HEADER	Accept	string	Zorunlu
KEY	invoice_id	string	Zorunlu
KEY	merchant_key	string	Zorunlu
KEY	order_id	string	Zorunlu
KEY	status	string	Zorunlu
KEY	hash_key	string	Zorunlu





**Başarısız Response:**

```
{
  "status_code": 31,
  "status_description": "Pending sale not found against order_id 166246147223278",
  "data": {
    "invoice_id": "f6a2a107-088f-4169-875f-8a6f30cf4ff2",
    "order_id": "166246147223278"
  }
}
```

**Başarılı Response:**

```
{
  "status_code": 100,
  "status_description": "Payment Successfully Completed",
  "data": {
    "order_no": "166254527653758",
    "order_id": "166254527653758",
    "invoice_id": "9a197ed9-cfaf-4456-b0fc-da74a95f461c",
    "status_code": 100,
    "status_description": "Payment Successfully Completed",
    "credit_card_no": "557023****7463",
    "transaction_type": "Auth",
    "payment_status": true,
    "payment_method": 1,
    "error_code": 100,
    "error": "Payment Successfully Completed",
    "auth_code": "287679",
    "status": "Completed",
    "hash_key":
      "c98d63c4ba1952c9:19e8:94Z0B48k__V5pCWIXlov3xbDKmM8beaNE3Lk4v3Yu5A1wfdPulFnDsJgFKJX
      +2gOqA2AfjcRExhYluWCF3A4LirelUkv26kozmlHc0gjbwIY="
  }
}
```

**Önemli Not:**

Ödeme için yanıt doğrulaması için lütfen White Label Smart 3D API ile Öde bölümünün yanıt bölümüne bakın.



## 8.Server Durum Kodları

Status Code	Description	Durum Kodu Açıklama
200	OK	Tamam
201	Created	Oluşturuldu
202	Accepted (Request accepted, and queued for execution)	Kabul edildi. (İstek kabul edildi ve yürütülmek üzere sıraya alındı)
400	Bad request	Hatalı istek
401	Authentication failure	Kimlik doğrulama hatası
403	Forbidden	Yasak
404	Resource not found	Kaynak bulunamadı
405	Method Not Allowed	Yönteme İzin Verilmiyor
409	Conflict	Çakışma
412	Precondition Failed	Önkoşul Başarısız
413	Request Entity Too Large	Talep Varlığı Çok Büyük
500	Internal Server Error	Dahili Sunucu Hatası
501	Not Implemented	Uygulanmadı
503	Service Unavailable	Hizmet Kullanılamıyor

## 9.Uygulama Durum Kodları

Status Code	Description	Durum Kodu Açıklama
1	Basic Validation	Temel Doğrulama
3	The invoice id already processed, The order with this invoice id 455323123214 is in the process, please wait or create order with new invoice id	Fatura kimliği zaten işlendi, Bu fatura numarası 455323123214 olan sipariş işlemde, lütfen bekleyin veya yeni fatura numarasıyla sipariş oluşturun
12	Items must be an array, Invalid Currency code, Invalid format of items	Öğeler bir dizi olmalıdır, Geçersiz Para Birimi kodu, Geçersiz öğe biçimi
13	The total of your item's price is not equal to the invoice total.	Ürününüzün bedelinin toplamı fatura toplamına eşit değildir.
14	Merchant Not Found !	Üye işyeri bulunamadı!
30	Invalid credentials	Geçersiz kimlik bilgileri
31	Transaction Not Found	İşlem Bulunamadı
32	Invalid Invoice Id, Order Not Completed	Geçersiz Fatura Kimliği, Sipariş Tamamlanmadı
33	Quantity must be integer value	Miktar tamsayı değeri olmalıdır
34	The payment integration method is not allowed. Please contact support.	Ödeme entegrasyonu yöntemine izin verilmiyor. Lütfen desteğe başvurun.
35	Credit Card Payment Option has not been defined	Kredi Kartıyla Ödeme Seçeneği Tanımlanmadı
36	POS Not Found	POS Bulunamadı
37	Merchant Pos Commission was not set. Please contact with service provider	İşyeri Pos Komisyonu belirlenmedi. Lütfen servis sağlayıcınızla iletişime geçin



38	The Merchant Commission was not set for this currency and payment method. Please try another payment method	Üye işyeri Komisyonu bu para birimi ve ödeme yöntemi için belirlenmemiştir. Lütfen başka bir ödeme yöntemi deneyin
39	Commission not found	Komisyon bulunamadı
40	Installment not found	Taksit bulunamadı
41	Order Failed	Sipariş Başarısız
41	Payment Failed	Ödeme Başarısız
42	Product price is less than commission, Product price is less than cost	Ürün fiyatı komisyondan az, Ürün fiyatı maliyetten az
43	Settlement date was not set	Ödeme tarihi belirlenmedi
44	This credit card is blocked.	Bu kredi kartı bloke edildi.
44	Merchant daily number of transaction limit is crossed	İşyeri günlük işlem sayısı limiti aşıldı
45	Merchant daily transaction amount limit is crossed	İşyeri günlük işlem tutarı limiti aşıldı
46	Merchant monthly number of transaction limit is crossed	Üye işyerinin aylık işlem sayısı aşıldı
47	Merchant monthly transaction amount limit is crossed	Üye işyerinin aylık işlem tutarı limiti aşıldı
48	POS currency does not matches with expected currency id,	POS para birimi beklenen para birimi kimliğiyle eşleşmiyor,
	Minimum transaction limit per transaction has been violated,	İşlem başına minimum işlem limiti ihlal edildi,
	Minimum transaction limit per transaction has been violated, POS currency does not matches with expected currency id	İşlem başına minimum işlem limiti ihlal edildi, POS para birimi beklenen para birimi kimliğiyle eşleşmiyor
49	Refund Failed, Total refund amount should not cross net amount, Sale Transaction not found, Insufficient balance, Non refund Transaction State, Please wait at least 30 seconds for another refund,	Geri Ödeme Başarısız, Toplam geri ödeme tutarı net tutarı geçmemeli, Satış işlemi bulunamadı, Yetersiz bakiye, Geri ödeme yapılmayan İşlem Durumu, Başka bir geri ödeme için lütfen en az 30 saniye bekleyin,
	Maximum transaction limit per transaction has been violated	İşlem başına maksimum işlem sınırı ihlal edildi
55	Recurring payment can not be installment sale,	Peşin ödemeli taksitli satış yapılamaz,
	Recurring number can not be empty,	Tekrarlanan numara boş olamaz,
	Recurring number must be an integer,	Tekrarlanan sayı bir tam sayı olmalıdır,
	Recurring number must be greater than 1,	Tekrarlanan sayı 1'den büyük olmalıdır,
	Recurring number should not be greater than 121,	Tekrarlanan sayı 121'den büyük olmamalıdır,
	The recurring cycle can not be empty,	Tekrarlanan döngü boş olamaz,
	The recurring cycle unit is not valid. It should be 'D', 'M' or 'Y',	Tekrarlanan döngü birimi geçerli değil. 'D', 'M' veya 'Y' olmalıdır,
	Recurring interval can not be empty,	Tekrarlanan aralık boş olamaz,
	Recurring interval must be an integer,	Yinelenen aralık bir tamsayı olmalıdır,
	Recurring interval must be greater than 0,	Tekrarlanan aralık 0'dan büyük olmalıdır,



	Recurring interval should not be greater than 99,	Tekrarlama aralığı 99'dan büyük olmamalıdır,
	Invalid recurring web hook key! Please check key name on Halkode ,	Geçersiz yinelenen web kancası anahtarı! Lütfen Halkode'daki anahtar adını kontrol edin,
	Recurring web hook key can not be empty. Please assign your web hook key on Halkode ,	Yinelenen web kancası anahtarı boş olamaz. Lütfen web kancası anahtarınızı Halkode'a atayın,
56	Invalid sale web hook key! Please check key name on Halkode	Geçersiz satış web kancası anahtarı! Lütfen Halkode'daki anahtar adını kontrol edin
60	The merchant is not allowed to perform transactions using this card.	Üye işyerinin bu kartı kullanarak işlem yapmasına izin verilmemektedir.
68	Total Amount mismatch with hash key,	Hash anahtarıyla Toplam Tutar uyumsuzluğu,
	Currency mismatch with hash_key,	Hash_key ile para birimi uyumsuzluğu,
	Merchant key mismatch with hash key,	Hash anahtarıyla satıcı anahtarı uyumsuzluğu,
	Installment number mismatch with hash key,	Hash anahtarı ile taksit numarası uyumsuzluğu,
	Pos Id mismatch with hash key,	Hash anahtarı ile Pos Id uyumsuzluğu,
	Invoice id mismatch with hash key	Hash anahtarıyla fatura kimliği uyumsuzluğu
69	Order is not processed yet	Sipariş henüz işleme alınmadı
70	Card Program Mismatch, Recurring Not Found	Kart Programı Uyumsuzluğu, Tekrarlayan Bulunamadı
71	Recurring plan update failed, Invalid Response or Unknown Error	Yinelenen plan güncellemesi başarısız oldu, Geçersiz Yanıt veya Bilinmeyen Hata
72	Failed to delete, Old payment could not be processed	Silinemedi, Eski ödeme işlenemedi
73	Recurring plan card add failed, Invalid Response or Unknown Error	Yinelenen plan kartı eklenemedi, Geçersiz Yanıt veya Bilinmeyen Hata
76	Foreign Cards are not Allowed for this Merchant.	Bu Satıcı için Yabancı Karta İzin Verilmez.
76	Foreign Cards are not Allowed for this Merchant.	Bu Satıcı için Yabancı Karta İzin Verilmez.
77	Foreign Card Commission is not set for this merchant	Bu üye işyeri için Yabancı Kart Komisyonu belirlenmedi
79	Merchant is not allowed to pay with card token	Satıcının kart jetonuyla ödeme yapmasına izin verilmiyor
80	Sub Merchant not found	Alt İşyeri bulunamadı
81	Currency conversion failed from API	API'den para birimi dönüştürme işlemi başarısız oldu
85	invalid character	Geçersiz karakter
86	Save Card Token failed	Kart Simgesini Kaydetme başarısız oldu
87	Invalid Token or customer number	Geçersiz Jeton veya müşteri numarası
88	The card token deletion failed	Kart belirtecinin silinmesi başarısız oldu
89	The card token update failed	Kart belirteci güncellemesi başarısız oldu
90	Invalid Hash key	Geçersiz Hash anahtarı
91	Hash key mismatched with merchant key	Hash anahtarı satıcı anahtarıyla eşleşmiyor
92	Hash key mismatched with customer number	Hash anahtarı müşteri numarasıyla eşleşmiyor
93	Hash key mismatched with card holder name	Karma anahtar kart sahibinin adıyla eşleşmiyor



94	Hash key mismatched with card number	Karma anahtarı kart numarasıyla eşleşmiyor
95	Hash key mismatched with expiry month	Hash anahtarı son kullanma ayıyla eşleşmiyor
96	Hash key mismatched with expiry year	Hash anahtarı son kullanma yılıyla eşleşmiyor
97	Hash key mismatched with card token	Karma anahtarı kart jetonuyla eşleşmiyor
99	Unknown Error	Bilinmeyen Hata
100	Success	Başarılı
102	Pos was not defined, Pos not found for this installment	Pos tanımlanmadı, bu taksit için Pos bulunamadı
103	PayBytoken commission was no setup	PayBytoken komisyonu ayarlanmadı
104	Refund transaction id must be unique	Geri ödeme işlem kimliği benzersiz olmalıdır
105	Transaction is not approved	İşlem onaylanmadı
106	Invalid Merchant Type	Geçersiz Satıcı Türü
107	Sent API OTP Unverified	Gönderilen API OTP Doğrulanmamış
108	Invalid Card Number	Geçersiz Kart Numarası
109	File Processing Error	Dosya İşleme Hatası
110	Partial Refund is not allowed for this transaction	Bu işlem için Kısmi İadeye izin verilmiyor
112	Partially Successful	Kısmen Başarılı
404	Cancel Url should not be empty	İptal URL'si boş olmamalıdır
113	There is no pre auth transaction found for this operation	Bu işlem için ön kimlik doğrulama işlemi bulunamadı