



HALK ÖDEME ENTEGRASYON TEKNİK DÖKÜMANI

HALK ELEKTRONİK PARA VE ÖDEME HİZMETLERİ VE A.Ş

İSTANBUL

1- Giriş	2
2- Neden HALKÖDE?	2
3- Başlamadan Önce	2
4- Ortamlar.....	2
5- İstekler	3
5. 1- Token Alma.....	3
5.1.1- Hash Hesaplama.....	4
5. 2- Purchase Link (Tek Kullanımlık DPL Oluşturma)	5
5. 3- PaySmart2D (Non-Secure Ödeme)	7
5. 4 - PaySmart3D	9
5.5- Refund API (İade).....	11
5.6 - Checkstatus (İşlem Durumu)	12
5. 7- Confirm Payment.....	13
5. 8- Get Installment (İş Yeri Aktif Taksitleri).....	14
5. 9- Get Pos (Taksit Bilgisi Alma)	15
5. 10- Merchant Commissions (Üye İşyeri Komisyon API)	16
5. 11 - Get Transactions.....	17
5. 12- All Transactions	18
6- Test Kartları.....	20
7- Canlı Ortama Geçiş	21
8- Destek	21
9- Server Durum Kodları	21
10- Uygulama Durum Kodları.....	22

1- Giriş

Bu doküman, HALKÖDE Sanal POS sisteminin entegrasyon sürecini adım adım açıklayan kapsamlı bir rehberdir.

HALKÖDE, işletmelerin ödeme işlemlerini kolaylaştırmak için geliştirilmiş bir Sanal POS çözümüdür. Bu rehber, HALKÖDE API'sini kullanarak ödeme sisteminizi nasıl entegre edeceğinizi ve temel işlemleri nasıl gerçekleştireceğinizi açıklar. Entegrasyon sürecinde ihtiyaç duyacağınız bilgiler, kod örnekleri ve dikkat edilmesi gereken noktalar burada yer almaktadır.

2- Neden HALKÖDE?

Hızlı ve Güvenli: Ödeme işlemleri saniyeler içinde tamamlanır ve üst düzey güvenlik protokolleriyle korunur.

Esnek Yapı: Farklı iş modellerine uyum sağlayacak şekilde özelleştirilebilir.

Kolay Entegrasyon: API dokümantasyonu ve örneklerle hızlı bir başlangıç yapabilirsiniz.

3- Başlamadan Önce

Entegrasyona başlamadan önce aşağıdaki gereksinimlere sahip olmanız gerekir:

HALKÖDE Hesabı: HALKÖDE müşteri paneli üzerinde bir üye işyeri hesabı oluşturmalsınız.

API Anahtarları: Entegrasyon için gerekli olan API Key ve Secret Key'i HALKÖDE müşteri panelinden alabilirsiniz.

4- Ortamlar

Test işlemleri için: <https://testapp.halkode.com.tr/ccpayment>

Canlı işlemler için: <https://app.halkode.com.tr/ccpayment>

5- İstekler

Bu bölümde, HALKÖDE müşteri panelinden alınan API Key ve Secret Key ile kullanılabilecek temel istekler detaylı bir şekilde açıklanmıştır. Her bir istek için metod, URL, içerik türü, Authorization durumu, parametreler, JSON istek ve yanıt örnekleri sırasıyla verilmiştir.

5. 1- Token Alma

API ile ödeme işlemlerine başlamadan önce bir token almanız gerekmektedir. Token, satıcıyı doğrulamak için diğer API'lerde kullanılacak bir oturum anahtarıdır. Ayrıca, satıcı için ayarlanan ödeme entegrasyon seçeneğini (2D veya 3D) de döndürür. Yanıt anahtarı is_3d'dir. is_3d'nin olası değerleri: 0, 1, 2 ve 4.

- 0:** Yalnızca Whitelabel 2D
- 1:** Whitelabel 2D veya 3D
- 2:** Yalnızca Whitelabel 3D
- 4:** Markalı ödeme çözümü

İstek Detayları

Metot: POST

URL: /api/token

Parametreler

Tür	Parametre	Veri Türü	Koşul	Açıklama
KEY	app_id	string	Zorunlu	İşyerine atanan benzersiz bir anahtar
KEY	app_secret	string	Zorunlu	İşyerine atanan gizli bir anahtar

JSON İstek Örneği

```
Json
{
  "app_id": "app_id numaranız",
  "app_secret": "app_secret numaranız"
}
```

JSON Yanıt Örneği

```
Json
{
  "status": "success",
  "token": "oluşturulan token numarası",
  "is_3d": 1,
  "message": "Token başarıyla oluşturuldu."
}
```

5.1.1- Hash Hesaplama

2D, 3D, İade, Ödeme Doğrulama işlemlerinde Ödeme işleminden sonra, son kullanıcı aşağıda verilen parametreler kullanılarak satıcı web sitesine yönlendirilir. Yanıt doğrulaması için bir hash_key parametresi gönderiyoruz. Satıcı, istek parametresini hash_key şifresi çözülmüş değerle karşılaştırabilir.

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": "Oder Failed",
    "transaction_status": "Failed",
    "order_id": "162435924998223",
    "transaction_id": "6jvQ7-l3kb-TC10-98950-220621",
    "message": "Oder 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": ""
}
```

5. 2- Purchase Link (Tek Kullanımlık DPL Oluşturma)

Tek kullanımlık bir Dinamik Ödeme Linki (DPL) oluşturmak için bu API kullanılır. Bu link, müşterilere ödeme yapmaları için gönderilebilir ve belirli bir süre veya kullanım limitiyle sınırlıdır.

İstek Detayları

Metot: POST

URL: /purchase/link

Authorization: Gerekli (Bearer token kullanılmalıdır)

Parametreler

Tür	Parametre	Veri Türü	Koşul	Açıklama
KEY	merchant_key	string	Zorunlu	Üye işyeri anahtarı
KEY	currency_code	string	Zorunlu	Para birimi (TRY)
KEY	invoice	string	Zorunlu	Eşsiz Fatura Numarası
KEY	name	string	Zorunlu	Müşteri Adı
KEY	surname	string	Zorunlu	Müşteri Soyadı
KEY	bill_address1	string	Opsiyonel	Fatura Adresi

KEY	bill_address2	string	Opsiyonel	Fatura Adresi
KEY	bill_city	string	Opsiyonel	Fatura Adresi şehir
KEY	bill_postcode	string	Opsiyonel	Fatura posta kodu
KEY	bill_state	string	Opsiyonel	Fatura ili
KEY	bill_country	string	Opsiyonel	Fatura ülkesi
KEY	bill_email	string	Opsiyonel	Faturaya tanımlı mail adresi
KEY	bill_phone	string	Opsiyonel	Fatura telefonu
KEY	ip	string	Opsiyonel	İp
KEY	transaction_type	string	Opsiyonel	transaction_type == "Auth" //işlem tutarı karttan anında düşülür. transaction_type == "Pre-Authorization" //işlem tutarı karttan daha sonra düşülecektir.

JSON İstek Örneği

```
{
  "merchant_key":"Müşteri Numaranız",
  "currency_code":"TRY",
  "invoice":{"
    "invoice_id\":"example-1375533702\","invoice_description\":"Testdescription\","total\":"10.00\","return_url\":"https://google.com.tr\","cancel_url\":"https://github.com.tr\","items\":[{"name\":"Item1\","price\":"100\","quantity\":"1\","description\":"Test\"}]
  },
  "name":"John",
  "surname":"Dao"
}
```

JSON Yanıt Örneği

```
{
  "status": true
  "status_code": 100,
  "success_message": "Success !",
  "link":"https://testapp.halkode.com.tr/ccpayment/makepayment/VP17459108361559304",
  "order_id": "VP17459108361559304"
}
```

5. 3- PaySmart2D (Non-Secure Ödeme)

Müşterilerin tek seferlik şifre girmeden ve güvenlik kontrolü olmadan ödeme yapmalarını sağlar. Bu yöntem, token almada is_3d değeri 0 veya 1 ise kullanılabilir.

İstek Detayları

Metot: POST

URL: /api/ paySmart2D

Authorization: Gerekli (Bearer token kullanılmalıdır)

Parametreler

Tür	Parametre	Veri Türü	Koşul	Açıklama
KEY	merchant_key	string	Zorunlu	Üye işyeri anahtarı
KEY	currency_code	string	Zorunlu	Para birimi (TRY)
KEY	invoice	string	Zorunlu	Eşsiz Fatura Numarası
KEY	name	string	Zorunlu	Müşteri Adı
KEY	surname	string	Zorunlu	Müşteri Soyadı
KEY	bill_address1	string	Opsiyonel	Fatura Adresi
KEY	bill_address2	string	Opsiyonel	Fatura Adresi
KEY	bill_city	string	Opsiyonel	Fatura Adresi şehir
KEY	bill_postcode	string	Opsiyonel	Fatura posta kodu
KEY	bill_state	string	Opsiyonel	Fatura ili
KEY	bill_country	string	Opsiyonel	Fatura ülkesi
KEY	bill_email	string	Opsiyonel	Faturaya tanımlı mail adresi
KEY	bill_phone	string	Opsiyonel	Fatura telefonu
KEY	ip	string	Opsiyonel	İp
KEY	transaction_type	string	Opsiyonel	transaction_type == "Auth" //işlem tutarı karttan anında düşülür. transaction_type == "Pre-Authorization" //işlem tutarı karttan daha sonra düşülecektir.
KEY	payment_completed_by	string	Opsiyonel	payment_completed_by == "app" //işlem tutarınınında gerçekleşecektir. payment_completed_by == "Merchant" //işlem tutarı üye iş yeri onayından sonra gerçekleşecektir. Default olarak app'dir.

JSON İstek Örneği

```
{  
  
  "cc_holder_name":"John Dao",  
  "cc_no":"4155141122223339",  
  "expiry_month":"12",  
  "expiry_year":"20",  
  "cvv":"555",  
  "currency_code":"TRY",  
  "installments_number": 1,  
  "invoice_id":"111112324115312",  
  "invoice_description":" INVOICE TEST DESCRIPTION",  
  "total":22.00,  
  "items":[{"name":"Item3","price":22,"quantity":1,"description":"item3 description"}],  
  "name" : "John",  
  "surname" : "Dao",  
  "merchant_key":" Eşsiz Müşteri Numaranız",  
  "hash_key" : "hesaplanan hash numaranız"  
}
```

JSON Yanıt Örneği

```
{  
  
  "status_code": 100,  
  "status_description": "Payment process successful",  
  "data": {  
    "order_no": "VP17459141322115179",  
    "order_id": "VP17459141322115179",  
    "invoice_id": "1111125324115312",  
    "credit_card_no": "415514****3339",  
    "transaction_type": "Auth",  
    "payment_status": 1,  
    "payment_method": 1,  
    "error_code": 100,  
    "error": "Transaction Successful",  
    "auth_code": 396050,  
    "merchant_commission": 0,  
    "user_commission": 0,  
    "merchant_commission_percentage": 0,  
    "merchant_commission_fixed": 0,  
    "hash_key": "Hesaplanan hash",  
    "original_bank_error_code": "",  
    "original_bank_error_description": ""  
  }  
}
```

5. 4 - PaySmart3D

3D güvenli ödeme almak için bu API kullanılır. Kullanıcı, banka doğrulama sayfasına yönlendirilir ve SMS koduyla ödeme doğrulanır.

İstek Detayları

Metot: POST (HTML form gönderimi olmalıdır, AJAX kullanılmamalıdır)

URL: /api/paySmart3D

Parametreler

Tür	Parametre	Veri Türü	Koşul	Açıklama
KEY	merchant_key	string	Zorunlu	Üye işyeri anahtarı
KEY	currency_code	string	Zorunlu	Para birimi (TRY)
KEY	invoice	string	Zorunlu	Eşsiz Fatura Numarası
KEY	name	string	Zorunlu	Müşteri Adı
KEY	surname	string	Zorunlu	Müşteri Soyadı
KEY	bill_address1	string	Opsiyonel	Fatura Adresi
KEY	bill_address2	string	Opsiyonel	Fatura Adresi
KEY	bill_city	string	Opsiyonel	Fatura Adresi şehir
KEY	bill_postcode	string	Opsiyonel	Fatura posta kodu
KEY	bill_state	string	Opsiyonel	Fatura ili
KEY	bill_country	string	Opsiyonel	Fatura ülkesi
KEY	bill_email	string	Opsiyonel	Faturaya tanımlı mail adresi
KEY	bill_phone	string	Opsiyonel	Fatura telefonu
KEY	ip	string	Opsiyonel	Ip
KEY	transaction_type	string	Opsiyonel	transaction_type == "Auth" //işlem tutarı karttan anında düşülür. transaction_type == "Pre-Authorization" //işlem tutarı karttan daha sonra düşülecektir.
KEY	payment_completed_by	string	Opsiyonel	payment_completed_by == "app" //işlem tutarının gerçekteşcektir. payment_completed_by == "Merchant" //işlem tutarı üye iş yeri onayından sonra gerçekteşcektir. Default olarak app'dir.

JSON İstek Örneği

```
{
  "cc_holder_name":"John Dao",
  "cc_no":"4155141122223339",
  "expiry_month":"12",
  "expiry_year":"20",
  "cvv":"555",
  "currency_code":"TRY",
  "installments_number": 1,
  "invoice_id":"111112324115312",
  "invoice_description":" INVOICE TEST DESCRIPTION",
  "total":22.00,
  "items":[{"name":"Item3","price":22,"quantity":1,"description":"item3 description"}],
  "name" : "John",
  "surname" : "Dao",
  "merchant_key":" Eşsiz Müşteri Numaranız",
  "hash_key" : "hesaplanan hash numaranız"
  "return_url": "http://merchant-domain.com/return",
  "cancel_url": "http://merchant-domain.com/cancel"
}
```

JSON Yanıt Örneği

```
{
  "status_code": 100,
  "status_description": "Payment process successful",
  "data": {
    "order_no": "VP17459141322115179",
    "order_id": "VP17459141322115179",
    "invoice_id": "1111125324115312",
    "credit_card_no": "415514****3339",
    "transaction_type": "Auth",
    "payment_status": 1,
    "payment_method": 1,
    "error_code": 100,
    "error": "Transaction Successful",
    "auth_code": 396050,
    "merchant_commission": 0,
    "user_commission": 0,
    "merchant_commission_percentage": 0,
    "merchant_commission_fixed": 0,
    "hash_key": "Hesaplanan hash",
    "original_bank_error_code": "",
    "original_bank_error_description": ""
  }
}
```

5.5- Refund API (İade)

Müşterilere iade yapmak için bu API kullanılır.

İstek Detayları

Metot: POST

URL: /api/refund

Authorization: Gerekli (Bearer token kullanılmalıdır)

Parametreler

Tür	Parametre	Veri Türü	Koşul	Açıklama
KEY	merchant_key	string	Zorunlu	Üye işyeri anahtarı
KEY	transaction_id	string	Zorunlu	Fatura kimliği
KEY	amount	float	Zorunlu	İade edilecek tutar
KEY	hash_key	string	Zorunlu	Güvenlik için oluşturulan hash anahtarı

JSON İstek Örneği

```
{
  "invoice_id": "Abcd124567",
  "amount": 8.00,
  "app_id": "benzersiz app_id",
  "app_secret": "benzersiz app_secret",
  "merchant_key": "benzersiz Merchant_key",
  "hash_key": "hesaplanan hash key"
}
```

JSON Yanıt Örneği

```
{
  "status_code": 100,
  "status_description": " success ",
  "order_no": "",
  "invoice_id": "Abcd124567",
  "ref_no": "",
  "ref_number": ""
}
```

5.6 - Checkstatus (İşlem Durumu)

Bir işlemin durumunu kontrol etmek için bu API kullanılır.

İstek Detayları

Metot: POST

URL: /api/checkstatus

Authorization: Gerekli (Bearer token kullanılmalıdır)

Parametreler

Tür	Parametre	Veri Türü	Koşul	Açıklama
KEY	merchant_key	string	Zorunlu	Üye işyeri anahtarı
KEY	invoice_id	string	Zorunlu	Fatura kimliği

JSON İstek Örneği

```
{  
  "merchant_key": "benzersiz üye iş yeri numarası",  
  "invoice_id": "33491162435928",  
}
```

JSON Yanıt Örneği

```
{  
  "status_code": 32,  
  "status_description": "Invalid invoice id",  
  "transaction_status": "Failed",  
  "order_id": "",  
  "transaction_id": "",  
  "message": "Invalid invoice id",  
  "reason": "No Request found for this invoice id",  
  "bank_status_code": "",  
  "bank_status_description": "",  
  "invoice_id": "Abcd12456",  
  "total_refunded_amount": 0,  
  "product_price": "0",  
  "transaction_amount": 0,  
  "ref_number": "",  
}
```

```

"transaction_type": "Auth",
"original_bank_error_code": "",
"original_bank_error_description": ""
}

```

5. 7- Confirm Payment

Ön provizyon (Pre-Authorization) işlemini tamamlamak ve bloke edilen tutarı karttan çekmek için bu API kullanılır.

İstek Detayları

Metot: POST

URL: /api/confirmPayment

Authorization: Gerekli (Bearer token kullanılmalıdır)

Parametreler

Tür	Parametre	Veri Türü	Koşul	Açıklama
KEY	merchant_key	string	Zorunlu	Üye işyeri anahtarı
KEY	invoice_id	string	Zorunlu	Fatura kimliği
KEY	hash_key	string	Zorunlu	Güvenlik için oluşturulan hash anahtarı

JSON İstek Örneği

```
{
  "total":"205.00",
  "invoice_id":"dz04557",
  "status":"1",
  "merchant_key":"benzersiz Merchant key",
  "hash_key":"hesaplanan hash"
}
```

JSON Yanıt Örneği

```
{
  "status_code": 100,
  "status_description": "Payment confirmed successfully",
  "transaction_id": "SXY0m-o3kb-TC10-98950-220621"
  "order_id": "",
  "invoice_id": "dz04557"
}
```

5. 8- Get Installment (İş Yeri Aktif Taksitleri)

İş yerinin aktif taksit seçeneklerini almak için bu API kullanılır.

Not:Banka kartında taksit olmaz ayrıca kredi kartında taksit olmayan kartlar ile test etmeyiniz.

İstek Detayları

Metot: POST

URL: /api/getinstallments

Authorization: Gerekli (Bearer token kullanılmalıdır)

Parametreler

Tür	Parametre	Veri Türü	Koşul	Açıklama
KEY	merchant_key	string	Zorunlu	Üye işyeri anahtarı

JSON İstek Örneği

```
{
  "merchant_key": "eşsiz üye iş yeri numarası"
}
```

JSON Yanıt Örneği

```
{
  "status_code": 100,
  "installments": [1, 3, 6, 9, 12],
  "message": "Aktif taksit bilgileri başarıyla alındı."
}
```

5. 9- Get Pos (Taksit Bilgisi Alma)

Kullanıcı kart numarasının ilk 6 hanesini (veya tarım kartları için tamamını) girdiğinde, taksit bilgisi almak için bu API kullanılır.

İstek Detayları

Metot: POST

URL: /api/getpos

Authorization: Gerekli (Bearer token kullanılmalıdır)

Parametreler

Tür	Parametre	Veri Türü	Koşul	Açıklama
KEY	credit_card	string	Zorunlu	Kart numarasının ilk 6 hanesi
KEY	amount	float	Zorunlu	Ödeme miktarı
KEY	currency_code	string	Zorunlu	Para birimi (TRY)
KEY	merchant_key	string	Zorunlu	Üye işyeri anahtarı

JSON Yanıt Örneği

```
{
  "status_code": 100,
  "status_description": "Successfull",
  "data": [
    {
      "pos_id": 1,
      "campaign_id": 0,
      "allocation_id": 0,
      "installments_number": 1,
      "card_type": "CREDIT CARD",
      "card_program": "WORLD",
      "card_scheme": "visa",

```



```

    "payable_amount": 100,
    "hash_key": "hesaplanan hash numarası",
    "amount_to_be_paid": "100.00",
    "currency_code": "TRY",
    "currency_id": 1,
    "title": "Single payment"
  }
}

```

5. 10- Merchant Commissions (Üye İşyeri Komisyon API)

Kart programına göre üye işyeri ve son kullanıcı komisyonlarını almak için bu API kullanılır.

İstek Detayları

Metot: GET

URL: /api/commissions

Authorization: Gerekli (Bearer token kullanılmalıdır)

Parametreler

Parametreler

Tür	Parametre	Veri Türü	Koşul	Açıklama
KEY	merchant_key	string	Zorunlu	Üye işyeri anahtarı

JSON İstek Örneği

```

{
  "currency_code": "TRY"
}

```

JSON Yanıt Örneği

```

{
  "status_code": 100,
  "status_description": "successful",
  "data": {
    "1": [
      {
        "card_program": "BANKKART_COMBO",
        "merchant_commission_percentage": "5.0000",
        "merchant_commission_fixed": "1.0000",
        "user_commission_percentage": 3,

```

```

    "user_commission_fixed": 1,
    "currency_code": "TRY",
    "installment": 1
  },
  {
    "card_program": "BONUS",
    "merchant_commission_percentage": "0.0000",
    "merchant_commission_fixed": "0.0000",
    "user_commission_percentage": 0,
    "user_commission_fixed": 0,
    "currency_code": "TRY",
    "installment": 1
  }
]
}
}

```

5. 11 - Get Transactions

Belirli bir işlem geçmişini almak için bu API kullanılır. (Not: Bu API, resmi dökümantasyonlarda açıkça belirtilmemiştir, ancak genel bir işlem geçmişi sorgulama mantığına dayanarak eklenmiştir.)

İstek Detayları

Metot: POST

URL: /api/getTransactions

Authorization: Gerekli (Bearer token kullanılmalıdır)

Parametreler

Tür	Parametre	Veri Türü	Koşul	Açıklama
KEY	merchant_key	string	Zorunlu	Üye işyeri anahtarı
KEY	start_date	string	Zorunlu	Başlangıç tarihi (YYYY-MM-DD)
KEY	end_date	string	Zorunlu	Bitiş tarihi (YYYY-MM-DD)

JSON İstek Örneği

```
{  
  "merchant_key":"eşsiz üye iş numarası",  
  "hash_key" : "hesaplanan hash key",  
  "date":"2024-04-17",  
  "invoiceid":"Abcd12456" // Optional  
}
```

JSON Yanıt Örneği

```
{  
  "status_code": 100,  
  "status_description": "Successfull",  
  "data": {  
    "transactions": [],  
    "paginate_data": {  
      "current_page": 1,  
      "total": 0,  
      "per_page": 10,  
      "last_page": 1,  
      "next_page_url": null,  
      "previous_page_url": null,  
      "current_page_url":  
        "https://testapp.halkode.com.tr/ccpayment/api/getTransactions?page=1",  
      "last_page_url":  
        "https://testapp.halkode.com.tr/ccpayment/api/getTransactions?page=1"  
    }  
  }  
}
```

5. 12- All Transactions

Tüm işlemleri almak için bu API kullanılır. (Not: Bu API, resmi dokümantasyonlarda açıkça belirtilmemiştir, ancak genel bir işlem listesi mantığına dayanarak eklenmiştir.)

İstek Detayları

Metot: POST

URL: /api/allTransactions

İçerik Türü: application/json

Authorization: Gerekli (Bearer token kullanılmalıdır)

Parametreler

Tür	Parametre	Veri Türü	Koşul	Açıklama
KEY	merchant_key	string	Zorunlu	Üye işyeri anahtarı

JSON İstek Örneği

```
{
  "merchant_key": " Üye işyeri anahtarı"
}
```

JSON Yanıt Örneği

```
{
  "status_code": 100,
  "status_description": "data fetched successfully",
  "data": {
    "current_page": 1,
    "data": [
      {
        "id": 3282,
        "payment_id": "qLrbl-WQ-TC10-92974158-290425",
        "invoice_id": "1111125324115312",
        "order_id": "VP17459141322115179",
        "remote_order_id": "174591413213574",
        "user_id": 1650,
        "end_user_id": 0,
        "user_name": "",
        "merchant_id": 92974158,
        "merchant_name": "",
        "transaction_state_id": 1,
        "purchase_id": 0,
        "gross": "22",
        "refund_request_amount": "0",
        "fee": "0",
        "net": "22",
        "pay_by_token_fee": "0",
        "cost": "0.22",
        "rolling_amount": "0",
        "product_price": "22",
        "user_commission": "0",

```

```

"merchant_commission": "0",
"gsm_number": "",
"settlement_date_bank": "2025-04-30 11:00:00",
"settlement_date_merchant": "2025-04-30 11:00:00",
"payment_type_id": 1,
"operator": "",
"issuer_name": "ZİRAAT BANKASI",
"card_program": "PARAF",
"card_issuer_name": "T. HALK BANKASI A.S.",
"pos_name": "DUMMY BANK",
"pos_id": 1,
"allocation_id": 0,
"url": "https://testapp.halkode.com.tr/ccpayment/api/alltransaction?page=2",
"label": "Next &raquo;",
"active": false
}
],
"next_page_url": "https://testapp.halkode.com.tr/ccpayment/api/alltransaction?page=2",
"path": "https://testapp.halkode.com.tr/ccpayment/api/alltransaction",
"per_page": 10,
"prev_page_url": "",
"to": 10,
"total": 90
}

```

6- Test Kartları

Test ortamında ödeme işlemlerini denemek için aşağıdaki kartları kullanabilirsiniz:

Visa: 4111 1111 1111 1111,

Son Kullanma: 12/25,

CVV: 123

- **MasterCard:** 5400 0000 0000 0001,

Son Kullanma: 12/25,

CVV: 456

7- Canlı Ortama Geçiş

Test işlemlerini tamamladıktan sonra canlı ortama geçmek için:

Halköde panelinden <https://app.halkode.com.tr/merchant/login> canlı API anahtarlarını alın.

Endpoint'i <https://app.halkode.com.tr> olarak güncelleyin.

Gerçek kartlarla test yaparak entegrasyonu doğrulayın.

8- Destek

Daha fazla bilgi veya destek için destek@halkode.com.tr adresi ile iletişime geçebilirsiniz.

9- 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

10- 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 kanca 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 Uyuşmazlığı, 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 anahtar 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 anahtar 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.