

Developer API

API Referansı

REST endpoint'leri, SignalR abonelikleri ve entegrasyon örnekleri.

Production: <https://api.tradewize.com.tr>

Entegrasyon testi: <https://testapi.tradewize.com.tr>

10 bölüm / 50 endpoint ve hub metodu

Eylül 2026 değişiklikleri

14 Eylül 2026 · ortam geçişi

Production adresi 14 Eylül 2026 Pazartesi itibarıyla <https://api.tradewize.com.tr> olarak değişir. Mevcut test ve production adresleri aynı tarihte devreden çıkarılır. Bu sürüm <https://testapi.tradewize.com.tr> üzerinde entegrasyon testlerine açıktır. Mevcut API key'leri kullanılabilir; yeni key oluşturulması gerekmez.

Test ortamı · gerçek hesap

Test ortamı gerçek piyasa verisine ve gerçek portföye bağlıdır; yalnız endpoint entegrasyon testleri içindir, sanal işlem ortamı değildir. Emir gönderimi ve bot güncellemeleri gerçek hesap üzerinde işlem yapar.

Son fiyat · yanıt ve hata davranışı

PAY/VİOP son fiyatları ayrı endpoint'lerden alınır. Compact yanıt korundu; details her sembol için yalnız price ve Unix milisaniye cinsinden timestamp içerir. Eksik/eski fiyat veya sembol listesi erişim sorunu nedeniyle toplu isteğin 503 dönmesi giderildi. Sonuçlar sembol bazında işlenir; kullanılabilir fiyatı olmayan sembol 0 döner. symbols en fazla 100 farklı sembol kabul eder; all=true ürünün tamamını seçer.

Grafik verileri

Kapanmış mumlar bars, oluşmakta olan mum live-bar, son fiyat olayları recent-ticks üzerinden alınır. bars; countBack, zaman aralığı, 1W/1M periyotları, cursor ve isPartial alanlarını destekler. daily-ohlc günlük fiyat alanlarıyla birlikte lastClose, lastUpdatedUtc ve isFinal içerir.

İstek limitleri

Güncel veri limitleri JWT + endpoint grubu bazındadır: PAY ve VİOP son fiyat grupları ayrı ayrı 1 istek/saniye; recent-ticks 1 istek/3 saniye; live-bar ve daily-ohlc ayrı ayrı 1 istek/5 saniye. Compact/details ve symbols/all=true aynı ürün sayacını paylaşır. Sembol sayısı maliyeti değiştirmez. Geçmiş mumların mevcut ağırlıklı ve saatlik kotaları korunmuştur.

LevelBot · kısmi güncelleme düzeltilmesi

PUT /api/v1/level-bots/{id} işleminde fiyat sınırları ve kâr alanlarının güncellenmesi düzeltildi. Gönderilmeyen bot ve detay alanları mevcut değerlerini korur. Gövdede id ile kullanılan eski PUT /api/v1/level-bots adresi desteklenmeye devam eder.

İçindekiler

Ortamlar ve başlangıç

Yetki kapsamaları

Rate limiting

Entegre kurumlar

Kimlik Doğrulama

- POST /oauth/token / API anahtarı ile token alma
- POST /oauth/token / Token yenileme

Hesaplar

- GET /api/v1/accounts / Hesapları listele
- POST /api/v1/accounts/login / Hesaba giriş (SMS doğrulamalı)
- POST /api/v1/accounts/positions / Pozisyonları getir
- POST /api/v1/accounts/daily-orders / Gün içi emirleri listele

Emirler

- POST /api/v1/orders / Emir gönder
- POST /api/v1/orders/chain / Zincir emir gönder
- POST /api/v1/orders/batch / Toplu emir gönder
- POST /api/v1/orders/update / Emir güncelle
- PATCH /api/v1/orders/patch / Emri kısmi güncelle
- POST /api/v1/orders/cancel / Emir iptal et
- POST /api/v1/orders/cancel/batch / Toplu emir iptali

OCO Emirler

- GET /api/v1/oco-orders / OCO emirleri listele
- POST /api/v1/oco-orders / OCO emir grubu oluştur
- POST /api/v1/oco-orders/register-pending / Bekleyen OCO bağlantısı kaydet
- DELETE /api/v1/oco-orders/{id} / OCO grubunu iptal et

Sistemler

- POST /api/v1/systems/list / Sistemleri listele

Level Bot'lar

- GET /api/v1/level-bots / Level Bot'ları listele
- GET /api/v1/level-bots/{id} / Level Bot getir
- POST /api/v1/level-bots / Level Bot oluştur
- PUT /api/v1/level-bots/{id} / Level Bot güncelle
- GET /api/v1/level-bots/{id}/pair-orders / Aktif çift emirler
- GET /api/v1/level-bots/{id}/transaction-history / İşlem geçmişi

TradingView Bot'lar

- GET /api/v1/trading-view-bots / TradingView Bot'ları listele
- GET /api/v1/trading-view-bots/{id} / TradingView Bot getir
- POST /api/v1/trading-view-bots / TradingView Bot oluştur
- PUT /api/v1/trading-view-bots / TradingView Bot güncelle
- GET /api/v1/trading-view-bots/{id}/pair-orders / Çift emirler

Listeler

- GET /api/v1/lists/indexes / Endeksleri listele
- GET /api/v1/lists/indexes/{code} / Endeks sembollerini getir
- GET /api/v1/lists/sectors / Sektörleri listele
- GET /api/v1/lists/markets / Pazarları listele
- GET /api/v1/lists/my / Kendi listelerim
- GET /api/v1/lists/my/{id} / Liste detayı
- POST /api/v1/lists/my / Liste oluştur
- PUT /api/v1/lists/my/{id} / Liste güncelle
- DELETE /api/v1/lists/my/{id} / Liste sil
- POST /api/v1/lists/my/{id}/add-symbols / Listeye sembol ekle
- DELETE /api/v1/lists/my/{id}/remove-symbols / Listedeki sembol çıkar

Piyasa & Fiyat Verisi

- GET /api/v1/market-data/last-price / PAY son fiyat - compact
- GET /api/v1/market-data/last-price/details / PAY son fiyat - details
- GET /api/v1/market-data/viop/last-price / VİOP son fiyat - compact
- GET /api/v1/market-data/viop/last-price/details / VİOP son fiyat - details
- GET /api/v1/market-data/daily-ohlc / Günlük OHLC
- GET /api/v1/market-data/bars / Kapanmış tarihsel barlar
- GET /api/v1/market-data/live-bar / Açık mum - live-bar
- GET /api/v1/market-data/recent-ticks / Son tick'ler - recent-ticks

Gerçek Zamanlı (SignalR / WebSocket)

- HUB /hubs/positions / SubscribeToPositions - emir değişikliklerine abone ol
- HUB /hubs/positions / SubscribeToPortfolio - portföy güncellemelerine abone ol

Ortamlar ve başlangıç

Production: <https://api.tradewize.com.tr>

Entegrasyon testi: <https://testapi.tradewize.com.tr>

14 Eylül 2026 Pazartesi itibarıyla yeni production adresi kullanılır; önceki test ve production adresleri devreden çıkarılır. Mevcut API key'leri kullanılabilir.

Test ortamı gerçek piyasa verisine ve gerçek portföye bağlıdır. Yalnız endpoint entegrasyon testleri içindir. Emir ve bot güncelleme istekleri gerçek hesap üzerinde işlem yapar.

Hızlı başlangıç

- Mevcut API anahtarıyla /oauth/token üzerinden access token alın.
- Sonraki isteklerde Authorization: Bearer <access_token> başlığını kullanın.
- Postman collection ve environment dosyaları WebApp dokümantasyon sayfasından indirilebilir. Environment varsayılan olarak entegrasyon test adresini kullanır.

Yetki kapsamları

- portfolio: Hesap, pozisyon ve portföy verilerini okuma.
- trade: Emir gönderme, iptal etme ve OCO emirleri.
- report: Bot, liste, sembol ve piyasa verilerini okuma / raporlama.
- manage: Aracı kurum girişi, bot ve liste oluşturma / güncelleme / silme.

Piyasa verisi okumalarında mevcut report yetkisi kullanılabilir; yeni veri endpoint'leri için ek scope gerekmez. İşlem ve hesap yetkileri korunur.

Rate limiting

Güncel veri limitleri JWT ve endpoint grubu bazındadır; sembol sayısından bağımsızdır. Limit aşımında 429 ve Retry-After yanıtı döner. Tarihsel veri kotaları ayrı uygulanır.

Son fiyat - PAY / ViOP

Kapsam: JWT + ürün grubu

Limit: 1 istek / saniye. Her üründe compact/details ve symbols/all=true ortak sayaç; PAY ile ViOP ayrı sayaç

- GET /api/v1/market-data/last-price
- GET /api/v1/market-data/last-price/details
- GET /api/v1/market-data/viop/last-price
- GET /api/v1/market-data/viop/last-price/details

Yanıt başlıkları: Retry-After, X-RateLimit-Retry-After, X-RateLimit-Limit, X-RateLimit-Remaining

Günlük OHLC

Kapsam: JWT + endpoint grubu

Limit: 1 istek / 5 saniye; live-bar ve diğer gruplardan ayrı sayaç

- GET /api/v1/market-data/daily-ohlc

Yanıt başlıkları: Retry-After, X-RateLimit-Retry-After, X-RateLimit-Limit, X-RateLimit-Remaining

Açık mum

Kapsam: JWT + endpoint grubu

Limit: 1 istek / 5 saniye; daily-ohlc ve diğer gruplardan ayrı sayaç

- GET /api/v1/market-data/live-bar

Yanıt başlıkları: Retry-After, X-RateLimit-Retry-After, X-RateLimit-Limit, X-RateLimit-Remaining

Son tick'ler

Kapsam: JWT + endpoint grubu

Limit: 1 istek / 3 saniye

- GET /api/v1/market-data/recent-ticks

Yanıt başlıkları: Retry-After, X-RateLimit-Retry-After, X-RateLimit-Limit, X-RateLimit-Remaining

Kapanmış tarihsel barlar - mevcut kotalar

Kapsam: Ağırlıklı kota API key başına; saatlik kota kullanıcı başına

Limit: 120 ağırlık birimi / dakika + 5.000 istek / saat. İstek ağırlığı 1 + ceil(tahmini bar sayısı / 500). Yeni saniyelik limit eklenmedi

- GET /api/v1/market-data/bars

Yanıt başlıkları: Retry-After (429), X-RateLimit-Limit, X-RateLimit-Remaining, X-RateLimit-Reset, X-RateLimit-Request-Weight, X-RateLimit-Global-*

Portföy ve gün içi emir sorguları

Kapsam: Kullanıcı + aracı kurum + hesap numarası + endpoint

Limit: Her endpoint için hesap numarası başına 1 istek / 3 saniye; endpoint sayaçları ayrıdır

- POST /api/v1/accounts/positions
- POST /api/v1/accounts/daily-orders

Yanıt başlıkları: Retry-After, X-RateLimit-Retry-After, X-RateLimit-Limit, X-RateLimit-Remaining

Emir iletimi

Kapsam: Kullanıcı + aracı kurum

Limit: Kurum bazında değişir; Marbaş Menkul için saatte 500 emir

- POST /api/v1/orders
- POST /api/v1/orders/chain
- POST /api/v1/orders/batch (her kalem 1 emir sayılır)
- POST /api/v1/oco-orders
- POST /api/v1/oco-orders/register-pending

Yanıt başlıkları: Retry-After, X-RateLimit-Retry-After, X-RateLimit-Limit, X-RateLimit-Remaining

Kurum bazlı limitler

Kurum	Portföy okuma	Emir iletimi
Marbaş Menkul	Hesap ve endpoint başına 3 saniyede 1 istek	Saatte 500 emir

Entegre kurumlar

- Marbaş Menkul: brokerageCode=MarbasMenkul

Kimlik Doğrulama

TradeWize Developer API, OAuth 2.0 Client Credentials akışını kullanır. API anahtarınızı bir JWT access token ile değiştirir, sonraki tüm çağrılarda bu token'ı gönderirsiniz.

POST /oauth/token

API anahtarı ile token alma

Kimlik doğrulama gerekmez

API anahtarınızı JWT access token ile değiştirir. Bu, akıştaki ilk adımdır; dönen access token'ı sonraki tüm isteklerde Authorization başlığında kullanırsınız.

API anahtarını gizli tutun: API anahtarınız hassas bir sırdır. Sürüm kontrolüne eklemeyin, istemci tarafı koda gömmeyin; ortam değişkeni veya güvenli bir sır yöneticisinde saklayın.

info: Access token 24 saat (86400 sn), refresh token 30 gün (2592000 sn) geçerlidir. Süre dolmadan refresh token ile yeni bir token seti alın.

Parametreler

Alan	Tip / kullanım	Açıklama
X-API-Key	header / string / zorunlu	Panelden aldığınız API anahtarı (tw_live_ ile başlar).
Content-Type	header / string / zorunlu	application/json olmalıdır.
grant_type	body / string / zorunlu	"api_key" değerini alır.

İstek gövdesi

```
{
  "grant_type": "api_key"
}
```

cURL

```
curl -X POST https://api.tradewize.com.tr/oauth/token \
-H "X-API-Key: tw_live_api_anahtarınız" \
-H "Content-Type: application/json" \
-d '{"grant_type": "api_key"}'
```

Örnek yanıt

```
{
  "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9...",
  "token_type": "Bearer",
  "expires_in": 86400,
  "refresh_token": "rf_PAKhXhWmSeeicx6bAb1pg1FroLm1ig7jyMOpe...",
  "refresh_expires_in": 2592000,
  "issued_at": "2026-07-06T12:00:00Z"
}
```

C# örneği

```
// TradeWizeClient içine ekleyin - API anahtarıyla ilk token setini alır.
public async Task<bool> AuthenticateAsync()
{
    var request = new HttpRequestMessage(HttpMethod.Post, "/oauth/token");
    request.Headers.Add("X-API-Key", _apiKey);
    request.Content = new StringContent(
        JsonSerializer.Serialize(new { grant_type = "api_key" }),
        Encoding.UTF8, "application/json");

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();

    var json = await response.Content.ReadAsStringAsync();
    StoreTokens(JsonSerializer.Deserialize<TokenResponse>(json));
    return true;
}

// Kullanım
var apiKey = Environment.GetEnvironmentVariable("TRADEWIZE_API_KEY")!;
var client = new TradeWizeClient(apiKey);
await client.AuthenticateAsync();
```

POST /oauth/token

Token yenileme

Kimlik doğrulama gerekmez

Elinizdeki refresh token ve API anahtarınız ile yeni bir access token seti alır. Yenileme isteğinde X-API-Key header'ı zorunludur.

Refresh token tek kullanımlıktır: Her yenilemede yeni bir refresh token döner; eskisi geçersiz olur. Aynı refresh token'ı tekrar kullanmak güvenlik olayı sayılır ve tüm token ailesi iptal edilir.

Parametreler

Alan	Tip / kullanım	Açıklama
X-API-Key	header / string / zorunlu	Token setinin bağlı olduğu Developer API anahtarı.
Content-Type	header / string / zorunlu	application/json olmalıdır.
grant_type	body / string / zorunlu	"refresh_token" değerini alır.
refresh_token	body / string / zorunlu	Önceki yanıtta dönen geçerli refresh token.

İstek gövdesi

```
{
  "grant_type": "refresh_token",
  "refresh_token": "rf_PAKhXhWmSeeicx6bAb1pg1FroLm1ig7jyM0pe..."
}
```

cURL

```
curl -X POST https://api.tradewize.com.tr/oauth/token \
-H "X-API-Key: tw_live_api_anahtarınız" \
-H "Content-Type: application/json" \
-d '{
  "grant_type": "refresh_token",
  "refresh_token": "rf_PAKhXhWmSeeicx6bAb1pg1FroLm1ig7jyM0pe..."
}'
```

Örnek yanıt

```
{
  "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9...",
  "token_type": "Bearer",
  "expires_in": 86400,
  "refresh_token": "rf_YENI_REFRESH_TOKEN...",
  "refresh_expires_in": 2592000,
  "issued_at": "2026-07-06T12:00:00Z"
}
```

C# örneği

```
// TradeWizeClient içine ekleyin - refresh token ile yeni token seti alır.
public async Task<bool> RefreshTokenAsync()
{
    if (string.IsNullOrEmpty(_refreshToken))
        return await AuthenticateAsync();

    var request = new HttpRequestMessage(HttpMethod.Post, "/oauth/token")
    {
        Content = new StringContent(
            JsonSerializer.Serialize(new
            {
                grant_type = "refresh_token",
                refresh_token = _refreshToken
            }),
            Encoding.UTF8, "application/json")
    };
    request.Headers.Add("X-API-Key", _apiKey);

    var response = await _httpClient.SendAsync(request);
    if (!response.IsSuccessStatusCode)
        return await AuthenticateAsync(); // refresh token süresi dolmuşsa API anahtarına düş

    var json = await response.Content.ReadAsStringAsync();
    StoreTokens(JsonSerializer.Deserialize<TokenResponse>(json));
    return true;
}
```

Hesaplar

Aracı kurum hesaplarınızı listeleyin, hesaba giriş yapın, pozisyonları ve gün içi emirleri okuyun. Tüm çağrılar JWT gerektirir; yanıtlar { success, message, data } zarfıyla döner.

GET /api/v1/accounts

Hesapları listele

JWT gerekli / Scope: portfolio

Yapılandırılmış tüm aracı kurumlardan kullanıcının hesaplarını toplayıp tek listede döndürür. Sorgular paralel yapılır.

info: Bir aracı kurum yanıt vermezse istek hataya düşmez; erişilebilen kurumlardan gelen hesaplar döndürülür. Diğer uçlarda hesabı tanımlamak için brokerageCode ve accountNumber alanlarını kullanın.

cURL

```
curl -X GET https://api.tradewize.com.tr/api/v1/accounts \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": [
    {
      "accountNumber": "77777",
      "name": "Ana Hesap",
      "description": null,
      "brokerageName": "TradeWize Yatırım Menkul Değerler",
      "brokerageCode": "TradeWiseYatirim",
      "isActive": true,
      "isDefault": true
    }
  ]
}
```

C# örneği

```
// Tüm aracı kurumlardaki hesapları listeler.
public async Task<string> GetAccountsAsync()
{
    var request = new HttpRequestMessage(HttpMethod.Get, "/api/v1/accounts");
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

POST /api/v1/accounts/login

Hesaba giriş (SMS doğrulamalı)

JWT gerekli / Scope: manage

Aracı kurum hesabına iki aşamalı giriş yapar. 1. aşamada SMS talep edilir, 2. aşamada gelen kod ile giriş tamamlanır.

Aşama 1 - SMS talebi: sessionId ve sms alanları OLMADAN gönderin. Yanıtta otp = 1 ve sessionId dolu geliyorsa SMS telefonunuza iletilmiştir.

Aşama 2 - SMS doğrulama: Aynı gövdeye 1. aşamadan gelen sessionId ile telefondaki sms kodunu ekleyip tekrar gönderin. Başarılıysa yanıt authenticated = true döner.

Parametreler

Alan	Tip / kullanım	Açıklama
brokerageCode	body / string / zorunlu	Aracı kurum kodu (örn. "TradeWiseYatirim"). Her iki aşamada gönderilir.
username	body / string / zorunlu	TC Kimlik No. Her iki aşamada gönderilir.
password	body / string / zorunlu	Hesap şifresi. Her iki aşamada gönderilir.
accountNumber	body / string / zorunlu	Aracı kurumdaki hesap numarası. Her iki aşamada gönderilir.
sessionId	body / string / opsiyonel	1. aşama yanıtından gelen oturum kimliği. Yalnızca 2. aşamada.
sms	body / string / opsiyonel	Telefona gelen SMS kodu. Yalnızca 2. aşamada.

İstek gövdesi

```
{
  "brokerageCode": "TradeWiseYatirim",
  "username": "12345678900",
  "password": "Sifre123",
  "accountNumber": "77777"
}
```

cURL

```
curl -X POST https://api.tradewize.com.tr/api/v1/accounts/login \
-H "Authorization: Bearer <access_token>" \
-H "Content-Type: application/json" \
-d '{
  "brokerageCode": "TradeWiseYatirim",
  "username": "12345678900",
  "password": "Sifre123",
  "accountNumber": "77777"
}'
```

Örnek yanıt

```
{
  "success": true,
  "message": "SMS gönderildi",
  "data": {
    "otp": 1,
    "sessionId": "41824250",
    "requiresSms": true,
    "authenticated": false,
    "additionalInfo": "SMS kodu telefonunuza gönderildi"
  }
}
```

C# örneği

```
// Aracı kurum hesabına SMS doğrulamalı giriş.
// Aşama 1: sessionId/sms boş → SMS gönderilir. Aşama 2: sessionId + sms eklenir.
public async Task<string> LoginAsync(
    string brokerageCode, string username, string password,
    string accountNumber, string? sessionId = null, string? sms = null)
{
    var request = new HttpRequestMessage(HttpMethod.Post, "/api/v1/accounts/login");
    request.Headers.Authorization = await GetAuthHeaderAsync();
    request.Content = new StringContent(
        JsonSerializer.Serialize(new
        {
            brokerageCode, username, password, accountNumber, sessionId, sms
        }),
        Encoding.UTF8, "application/json");

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}

// Kullanım
await client.LoginAsync("TradeWiseYatirim", "12345678900", "Sifre123", "77777"); // Aşama 1
await client.LoginAsync("TradeWiseYatirim", "12345678900", "Sifre123", "77777", "41824250", "642529"); // Aşama 2
```

POST /api/v1/accounts/positions

Pozisyonları getir

JWT gerekli / Scope: portfolio

Hesabın pozisyonlarını, nakit değerlerini ve portföy özetini döndürür. includeOrders true ise bekleyen/tamamlanan emir listeleri de eklenir.

Hız sınırı: Her hesap numarası için bu endpoint 3 saniyede 1 kez çağrılabilir. Farklı hesap numaraları ve günlük emir endpoint'i ayrı sayaç kullanır. Aşırsa 429 yanıtı ve Retry-After başlığı döner.

Parametreler

Alan	Tip / kullanım	Açıklama
brokerageCode	body / string / zorunlu	Aracı kurum kodu.
accountNumber	body / string / zorunlu	Hesap numarası.
includeOrders	body / boolean / opsiyonel	true = pozisyonlar + emir listeleri, false = yalnızca pozisyonlar.

İstek gövdesi

```
{
  "brokerageCode": "TradeWiseYatirim",
  "accountNumber": "77777",
  "includeOrders": false
}
```

cURL

```
curl -X POST https://api.tradewize.com.tr/api/v1/accounts/positions \
-H "Authorization: Bearer <access_token>" \
-H "Content-Type: application/json" \
-d '{"brokerageCode": "TradeWiseYatirim", "accountNumber": "77777", "includeOrders": false}'
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": {
    "accountNumber": "77777",
    "dateTime": "2026-07-06T14:32:00Z",
    "accountBalance": 152340.50,
    "transactionLimit": 48000.00,
    "cashAmount": 12500.00,
    "t1NetCashAmount": 12500.00,
    "t2NetCashAmount": 12500.00,
    "overall": 152340.50,
    "stockOverall": 139840.50,
    "totalProfit": 8420.75,
    "positions": [
      {
        "symbol": "THYAO",
        "lot": 100,
        "sellable": 100,
        "profit": 3250.00,
        "bloke": 0,
        "dateTime": "2026-07-06T14:32:00Z"
      }
    ],
    "waitingList": [],
    "completedList": [
      {
        "symbol": "THYAO",
        "buySell": 1,
        "quantity": 100,
        "orderStatus": "Completed",
        "orderNo": "305344701",
        "description": "",
        "executedPrice": 310.50,
        "executedAt": "2026-07-06T14:32:11"
      }
    ],
    "pendingList": [],
    "cancelledList": []
  }
}
```

C# örneği

```
// Hesap pozisyonlarını ve portföy özetini getirir.
// Aynı accountNumber için bu endpoint 3 saniyede 1 kez çağrılabilir.
public async Task<string> GetPositionsAsync(
    string brokerageCode, string accountNumber, bool includeOrders = false)
{
    var request = new HttpRequestMessage(HttpMethod.Post, "/api/v1/accounts/positions");
    request.Headers.Authorization = await GetAuthHeaderAsync();
    request.Content = new StringContent(
        JsonSerializer.Serialize(new { brokerageCode, accountNumber, includeOrders }),
        Encoding.UTF8, "application/json");

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

POST /api/v1/accounts/daily-orders

Gün içi emirleri listele

JWT gerekli / Scope: portfolio

Hesapta o işlem günü verilen tüm emirleri (bekleyen ve tamamlanan) listeler.

info: buySell: 1 = Alış, 2 = Satış. orderStatus değerleri: Pending, Waiting, Completed, Canceled, MarketSell.

info: executedPrice: emir gerçekleştiğinde oluşan ortalama işlem fiyatı (henüz gerçekleşmemişse 0). executedAt: emrin gerçekleştiği an (saniye hassasiyetinde; henüz gerçekleşmemişse null).

Hız sınırı: Her hesap numarası için bu endpoint 3 saniyede 1 kez çağrılabilir. Farklı hesap numaraları ve pozisyon endpoint'i ayrı sayaç kullanır. Aşılırsa 429 yanıtı ve Retry-After başlığı döner.

Parametreler

Alan	Tip / kullanım	Açıklama
brokerageCode	body / string / zorunlu	Aracı kurum kodu.
accountNumber	body / string / zorunlu	Hesap numarası.

İstek gövdesi

```
{
  "brokerageCode": "TradeWiseYatirim",
  "accountNumber": "77777"
}
```

cURL

```
curl -X POST https://api.tradewize.com.tr/api/v1/accounts/daily-orders \
-H "Authorization: Bearer <access_token>" \
-H "Content-Type: application/json" \
-d '{"brokerageCode": "TradeWiseYatirim", "accountNumber": "77777"}'
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": [
    {
      "symbol": "THYAO",
      "buySell": 1,
      "quantity": 100,
      "orderStatus": "Completed",
      "orderNo": "305344701",
      "description": "",
      "executedPrice": 310.50,
      "executedAt": "2026-07-06T14:32:11"
    },
    {
      "symbol": "EREGL",
      "buySell": 2,
      "quantity": 50,
      "orderStatus": "Pending",
      "orderNo": "305344702",
      "description": "",
      "executedPrice": 0,
      "executedAt": null
    }
  ]
}
```

C# örneği

```
// Hesabın gün içi emirlerini listeler.
// Aynı accountNumber için bu endpoint 3 saniyede 1 kez çağrılabilir.
public async Task<string> GetDailyOrdersAsync(string brokerageCode, string accountNumber)
{
    var request = new HttpRequestMessage(HttpMethod.Post, "/api/v1/accounts/daily-orders");
    request.Headers.Authorization = await GetAuthHeaderAsync();
    request.Content = new StringContent(
        JsonSerializer.Serialize(new { brokerageCode, accountNumber },
            Encoding.UTF8, "application/json");

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

Emirler

Emir gönderin, zincir ve toplu emir verin, bekleyen emirleri güncelleyin veya iptal edin. Tüm uçlar trade kapsamı gerektirir. Ortak alanlar: buySell (1 = Alış, 2 = Satış), orderType (1 = Piyasa, 2 = Limit).

POST /api/v1/orders

Emir gönder

JWT gerekli / Scope: trade

Aracı kuruma tek bir emir iletir.

info: orderStatus değerleri: -1 UnProcessedSell, 0 UnProcessed, 1 Pending, 2 Waiting, 3 Completed, 4 Canceled, 5 MarketSell.

Rate limit: Emir iletimi kullanıcı + aracı kurum başına kurum kotasına tabidir (Marbaş Menkul: saatte 500 emir). Tüm limitler için Rate limiting bölümüne bakın.

Parametreler

Alan	Tip / kullanım	Açıklama
brokerageCode	body / string / zorunlu	Aracı kurum kodu.
accountNumber	body / string / zorunlu	Hesap numarası.
symbol	body / string / zorunlu	İşlem sembolü (örn. "FROTO").
quantity	body / integer / zorunlu	Emir miktarı (lot).
buySell	body / integer / zorunlu	1 = Alış, 2 = Satış.
orderType	body / integer / zorunlu	1 = Piyasa (price 0), 2 = Limit.
price	body / decimal / opsiyonel	Emir fiyatı. Limit emirde zorunlu, piyasa emirde 0.

İstek gövdesi

```
{
  "brokerageCode": "TradeWiseYatirim",
  "accountNumber": "77777",
  "symbol": "FROTO",
  "quantity": 1,
  "buySell": 1,
  "orderType": 2,
  "price": 101.50
}
```

cURL

```
curl -X POST https://api.tradewize.com.tr/api/v1/orders \
-H "Authorization: Bearer <access_token>" \
-H "Content-Type: application/json" \
-d '{
  "brokerageCode": "TradeWiseYatirim",
  "accountNumber": "77777",
  "symbol": "FROTO",
  "quantity": 1,
  "buySell": 1,
  "orderType": 2,
  "price": 101.50
}'
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": {
    "symbol": "FROTO",
    "quantity": 1,
    "buySell": 1,
    "orderNo": "12345678",
    "orderStatus": 1
  }
}
```

C# örneği

```
// Tek emir gönderir. orderType: 1 = Piyasa (price 0), 2 = Limit.
public async Task<string> SendOrderAsync(
    string brokerageCode, string accountNumber, string symbol,
    int quantity, int buySell, int orderType, decimal price)
{
    var request = new HttpRequestMessage(HttpMethod.Post, "/api/v1/orders");
    request.Headers.Authorization = await GetAuthHeaderAsync();
    request.Content = new StringContent(
        JsonSerializer.Serialize(new
        {
            brokerageCode, accountNumber, symbol, quantity, buySell, orderType, price
        }),
        Encoding.UTF8, "application/json");

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}

// Kullanım (limit alış)
await client.SendOrderAsync("TradeWiseYatirim", "77777", "FROTO", 1, 1, 2, 101.50m);
```

POST /api/v1/orders/chain

Zincir emir gönder

JWT gerekli / Scope: trade

Var olan bir emre bağlı (referans emir gerçekleşince tetiklenen) zincir emir iletir.

Rate limit: Emir iletimi kullanıcı + aracı kurum başına kurum kotasına tabidir (Marbaş Menkul: saatte 500 emir). Tüm limitler için Rate limiting bölümüne bakın.

Parametreler

Alan	Tip / kullanım	Açıklama
brokerageCode	body / string / zorunlu	Aracı kurum kodu.
accountNumber	body / string / zorunlu	Hesap numarası.
symbol	body / string / zorunlu	İşlem sembolü.
quantity	body / integer / zorunlu	Emir miktarı (lot).
buySell	body / integer / zorunlu	1 = Alış, 2 = Satış.
orderType	body / integer / zorunlu	1 = Piyasa, 2 = Limit.
price	body / decimal / opsiyonel	Emir fiyatı (limit emirde zorunlu).
referenceOrderNo	body / string / zorunlu	Bağlanılacak emrin numarası.

İstek gövdesi

```
{
  "brokerageCode": "TradeWiseYatirim",
  "accountNumber": "77777",
  "symbol": "ADESE",
  "quantity": 100,
  "buySell": 1,
  "orderType": 2,
  "price": 1.50,
  "referenceOrderNo": "305344701"
}
```

cURL

```
curl -X POST https://api.tradewize.com.tr/api/v1/orders/chain \
-H "Authorization: Bearer <access_token>" \
-H "Content-Type: application/json" \
-d '{
  "brokerageCode": "TradeWiseYatirim",
  "accountNumber": "77777",
  "symbol": "ADESE",
  "quantity": 100,
  "buySell": 1,
  "orderType": 2,
  "price": 1.50,
  "referenceOrderNo": "305344701"
}'
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": {
    "symbol": "ADESE",
    "quantity": 100,
    "buySell": 1,
    "orderNo": "12345679",
    "orderStatus": 1
  }
}
```

C# örneği

```
// Referans emre bağlı zincir emir gönderir.
public async Task<string> SendChainOrderAsync(
    string brokerageCode, string accountNumber, string symbol,
    int quantity, int buySell, int orderType, decimal price, string referenceOrderNo)
{
    var request = new HttpRequestMessage(HttpMethod.Post, "/api/v1/orders/chain");
    request.Headers.Authorization = await GetAuthHeaderAsync();
    request.Content = new StringContent(
        JsonSerializer.Serialize(new
        {
            brokerageCode, accountNumber, symbol, quantity,
            buySell, orderType, price, referenceOrderNo
        }),
        Encoding.UTF8, "application/json");

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

POST /api/v1/orders/batch

Toplu emir gönder

JWT gerekli / Scope: trade

Tek istekte birden çok emir iletir. Aracı kurum ve hesap bir kez girilir; her emir için yalnızca emre özgü alanlar tekrarlanır.

Rate limit: Batch isteğinde her emir kalemi kurumun kota penceresine ayrı sayılır (Marbaş Menkul: saatte 500 emir). Tüm limitler için Rate limiting bölümüne bakın.

Parametreler

Alan	Tip / kullanım	Açıklama
brokerageCode	body / string / zorunlu	Aracı kurum kodu (tüm emirler için ortak).
accountNumber	body / string / zorunlu	Hesap numarası (ortak).
items	body / array / zorunlu	Emir kalemleri: symbol, quantity, buySell, orderType, price.

İstek gövdesi

```
{
  "brokerageCode": "TradewiseYatirim",
  "accountNumber": "77777",
  "items": [
    { "symbol": "FROTO", "quantity": 10, "buySell": 1, "orderType": 2, "price": 101.50 },
    { "symbol": "THYAO", "quantity": 5, "buySell": 1, "orderType": 2, "price": 250.00 },
    { "symbol": "EREGL", "quantity": 20, "buySell": 2, "orderType": 1, "price": 0 }
  ]
}
```

cURL

```
curl -X POST https://api.tradewize.com.tr/api/v1/orders/batch \
-H "Authorization: Bearer <access_token>" \
-H "Content-Type: application/json" \
-d '{
  "brokerageCode": "TradewiseYatirim",
  "accountNumber": "77777",
  "items": [
    { "symbol": "FROTO", "quantity": 10, "buySell": 1, "orderType": 2, "price": 101.50 },
    { "symbol": "EREGL", "quantity": 20, "buySell": 2, "orderType": 1, "price": 0 }
  ]
}'
```

Örnek yanıt

```
{
  "success": true,
  "message": "2 orders processed successfully",
  "data": [
    { "symbol": "FROTO", "quantity": 10, "buySell": 1, "orderNo": "12345680", "orderStatus": 1 },
    { "symbol": "EREGL", "quantity": 20, "buySell": 2, "orderNo": "12345681", "orderStatus": 1 }
  ]
}
```

C# örneği

```
// Toplu emir gönderir.
public record BatchOrderItem(string Symbol, int Quantity, int BuySell, int OrderType, decimal Price);

public async Task<string> SendOrderBatchAsync(
    string brokerageCode, string accountNumber, IEnumerable<BatchOrderItem> items)
{
    var request = new HttpRequestMessage(HttpMethod.Post, "/api/v1/orders/batch");
    request.Headers.Authorization = await GetAuthHeaderAsync();
    request.Content = new StringContent(
        JsonSerializer.Serialize(new { brokerageCode, accountNumber, items }),
        Encoding.UTF8, "application/json");

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

POST /api/v1/orders/update

Emir güncelle

JWT gerekli / Scope: trade

Aracı kuruma iletilmiş bekleyen bir emrin miktarını, fiyatını veya emir tipini günceller. Bu işlem aracı kurumun emir güncelleme desteğine bağlıdır.

Piyasa ve limit emri: orderType=1 piyasa emridir ve price backend tarafından 0 olarak normalize edilir. orderType=2 limit emridir; bu durumda price 0'dan büyük olmalıdır.

Aracı kurum desteği: Emir güncelleme özelliği aracı kuruma bağlıdır. Kurum güncellemeyi desteklemiyorsa yanıt gövdesinde success=false döner; bu durumda emri iptal edip yeniden göndermeniz gerekir.

Parametreler

Alan	Tip / kullanım	Açıklama
brokerageCode	body / string / zorunlu	Emrin bulunduğu hesabın aracı kurum kodu.
accountNumber	body / string / zorunlu	Emrin ait olduğu aracı kurum hesap numarası.
orderNo	body / string / zorunlu	Gün içi emir listesinden alınan broker emir numarası.
symbol	body / string / zorunlu	Emrin BIST sembolü (ör. FROTO).
quantity	body / integer / zorunlu	Emrin güncelleneceği yeni toplam lot miktarı.
previousQuantity	body / integer / zorunlu	Emrin aracı kurumdaki güncelleme öncesi toplam lot miktarı.
price	body / decimal / zorunlu	Yeni fiyat. Piyasa emrinde 0, limit emrinde 0'dan büyük olmalıdır.
orderType	body / integer / zorunlu	1 = Piyasa, 2 = Limit.

İstek gövdesi

```
{
  "brokerageCode": "TradeWiseYatirim",
  "accountNumber": "77777",
  "orderNo": "305344701",
  "symbol": "FROTO",
  "quantity": 120,
  "previousQuantity": 100,
  "price": 104.50,
  "orderType": 2
}
```

cURL

```
curl -X POST https://api.tradewize.com.tr/api/v1/orders/update \
-H "Authorization: Bearer <access_token>" \
-H "Content-Type: application/json" \
-d '{
  "brokerageCode": "TradeWiseYatirim",
  "accountNumber": "77777",
  "orderNo": "305344701",
  "symbol": "FROTO",
  "quantity": 120,
  "previousQuantity": 100,
  "price": 104.50,
  "orderType": 2
}'
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": true
}
```

C# örneği

```
// Bekleyen bir emri günceller.
public async Task<string> UpdateOrderAsync(
    string brokerageCode, string accountNumber, string orderNo, string symbol,
    int quantity, int previousQuantity, decimal price, int orderType)
{
    var request = new HttpRequestMessage(HttpMethod.Post, "/api/v1/orders/update");
    request.Headers.Authorization = await GetAuthHeaderAsync();
    request.Content = new StringContent(
        JsonSerializer.Serialize(new {
            brokerageCode, accountNumber, orderNo, symbol,
            quantity, previousQuantity, price, orderType
        })),
        Encoding.UTF8, "application/json");

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

PATCH /api/v1/orders/patch

Emri kısmi güncelle

JWT gerekli / Scope: trade

Emrin yalnızca gönderilen yeni fiyat, yeni miktar veya yeni emir tipi alanlarını değiştirir. Gönderilmeyen değerler Robot.Api tarafından aracı kurumdan okunup korunur.

Kısmi güncelleme: newQuantity, newPrice ve newOrderType alanlarından en az biri gönderilmelidir. Gönderilmeyen emir alanları aracı kurumdan okunup aynı değerle yeniden iletilir.

Mevcut miktar kontrolü: quantity yeni miktar değildir. İstemcinin son gördüğü mevcut miktardır. Brokerdaki miktar değişmişse istek reddedilir; güncel emri okuyup yeniden deneyin.

Parametreler

Alan	Tip / kullanım	Açıklama
brokerageCode	body / string / zorunlu	Emrin bulunduğu hesabın aracı kurum kodu.
accountNumber	body / string / zorunlu	Emrin ait olduğu aracı kurum hesap numarası.
orderNo	body / string / zorunlu	Gün içi emir listesinden alınan broker emir numarası.
quantity	body / integer / zorunlu	İstemcinin son gördüğü mevcut toplam miktar; eşzamanlılık kontrolü için kullanılır.
newQuantity	body / integer / opsiyonel	Yeni toplam miktar. Miktarı korumak için göndermeyin.
newPrice	body / decimal / opsiyonel	Yeni fiyat. Fiyatı korumak için göndermeyin.
newOrderType	body / integer / opsiyonel	Yeni emir tipi: 1 = Piyasa, 2 = Limit. Tipi korumak için göndermeyin.

İstek gövdesi

```
{
  "brokerageCode": "MARBAS",
  "accountNumber": "77777",
  "orderNo": "305344701",
  "quantity": 100,
  "newPrice": 123.00
}
```

cURL

```
curl -X PATCH https://api.tradewize.com.tr/api/v1/orders/patch -H "Authorization: Bearer <access_token>" -H
"Content-Type: application/json" -d '{
  "brokerageCode": "MARBAS",
  "accountNumber": "77777",
  "orderNo": "305344701",
  "quantity": 100,
  "newPrice": 123.00
}'
```

Örnek yanıt

```
{
  "success": true,
  "message": "Order patched.",
  "data": true
}
```

C# örneği

```
// Yalnızca fiyatı değiştirir.
var request = new HttpRequestMessage(HttpMethod.Patch, "/api/v1/orders/patch");
request.Headers.Authorization = await GetAuthHeaderAsync();
request.Content = new StringContent(
    JsonSerializer.Serialize(new {
        brokerageCode = "MARBAS",
        accountNumber = "77777",
        orderNo = "305344701",
        quantity = 100,
        newPrice = 123.00m
    })),
    Encoding.UTF8, "application/json");

var response = await _httpClient.SendAsync(request);
response.EnsureSuccessStatusCode();
```

POST /api/v1/orders/cancel

Emir iptal et

JWT gerekli / Scope: trade

Var olan bir emri iptal eder.

Parametreler

Alan	Tip / kullanım	Açıklama
brokerageCode	body / string / zorunlu	Aracı kurum kodu.
accountNumber	body / string / zorunlu	Hesap numarası.
orderNo	body / string / zorunlu	İptal edilecek emrin numarası.
symbol	body / string / zorunlu	Emrin sembolü.

İstek gövdesi

```
{
  "brokerageCode": "TradeWiseYatirim",
  "accountNumber": "77777",
  "orderNo": "12345678",
  "symbol": "FROTO"
}
```

cURL

```
curl -X POST https://api.tradewize.com.tr/api/v1/orders/cancel \
-H "Authorization: Bearer <access_token>" \
-H "Content-Type: application/json" \
-d '{"brokerageCode": "TradeWiseYatirim", "accountNumber": "77777", "orderNo": "12345678", "symbol": "FROTO"}'
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": true
}
```

C# örneği

```
// Emir iptal eder.
public async Task<string> CancelOrderAsync(
    string brokerageCode, string accountNumber, string orderNo, string symbol)
{
    var request = new HttpRequestMessage(HttpMethod.Post, "/api/v1/orders/cancel");
    request.Headers.Authorization = await GetAuthHeaderAsync();
    request.Content = new StringContent(
        JsonSerializer.Serialize(new { brokerageCode, accountNumber, orderNo, symbol }),
        Encoding.UTF8, "application/json");

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

POST /api/v1/orders/cancel/batch

Toplu emir iptali

JWT gerekli / Scope: trade

Tek istekte birden çok emri iptal eder. Aracı kurum ve hesap bir kez girilir; her iptal için yalnızca orderNo ve symbol tekrarlanır.

Parametreler

Alan	Tip / kullanım	Açıklama
brokerageCode	body / string / zorunlu	Aracı kurum kodu (ortak).
accountNumber	body / string / zorunlu	Hesap numarası (ortak).
items	body / array / zorunlu	İptal kalemleri: orderNo, symbol.

İstek gövdesi

```
{
  "brokerageCode": "TradeWiseYatirim",
  "accountNumber": "77777",
  "items": [
    { "orderNo": "12345678", "symbol": "FROTO" },
    { "orderNo": "12345679", "symbol": "THYAO" }
  ]
}
```

cURL

```
curl -X POST https://api.tradewize.com.tr/api/v1/orders/cancel/batch \
-H "Authorization: Bearer <access_token>" \
-H "Content-Type: application/json" \
-d '{
  "brokerageCode": "TradewiseYatirim",
  "accountNumber": "77777",
  "items": [
    { "orderNo": "12345678", "symbol": "FROTO" },
    { "orderNo": "12345679", "symbol": "THYAO" }
  ]
}'
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": true
}
```

C# örneği

```
// Toplu emir iptali.
public record CancelBatchItem(string OrderNo, string Symbol);

public async Task<string> CancelOrderBatchAsync(
    string brokerageCode, string accountNumber, IEnumerable<CancelBatchItem> items)
{
    var request = new HttpRequestMessage(HttpMethod.Post, "/api/v1/orders/cancel/batch");
    request.Headers.Authorization = await GetAuthHeaderAsync();
    request.Content = new StringContent(
        JsonSerializer.Serialize(new { brokerageCode, accountNumber, items }),
        Encoding.UTF8, "application/json");

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

OCO Emirler

OCO (One-Cancels-the-Other): 1 Stop-Loss + kademeli Take-Profit emirlerinden oluşan sentetik gruplar. Borsaya gönderilmez, TradeWize tarafında saklanır; fiyat TP'ye ulaşınca o kademe, SL'ye düşünce tüm grup için piyasa satış tetiklenir. Fiyatlar BIST kademe aralığına uymalıdır.

GET /api/v1/oco-orders

OCO emirleri listele

JWT gerekli / Scope: portfolio

Hesaba ait tüm OCO satırlarını döndürür. Her TP ve SL ayrı bir satır olarak gelir; aynı groupId değerine sahip satırlar tek grubu oluşturur.

info: orderType: 1 = Take-Profit, 2 = Stop-Loss. status: 1 = Aktif, 2 = Tetiklendi, 3 = Tetikleme başarısız, 4 = İptal edildi.

Parametreler

Alan	Tip / kullanım	Açıklama
brokerageCode	query / string / zorunlu	Aracı kurum kodu.
accountNumber	query / string / zorunlu	Hesap numarası.

cURL

```
curl -X GET "https://api.tradewize.com.tr/api/v1/oco-orders?brokerageCode=TradeWiseYatirim&accountNumber=77777" \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": [
    {
      "id": 1001,
      "groupId": 1001,
      "accountNumber": "77777",
      "symbol": "EREGL",
      "orderType": 2,
      "quantity": 100,
      "price": 21.00,
      "status": 1,
      "triggeredPrice": null,
      "triggeredDate": null,
      "dueDate": null,
      "notes": null,
      "createdDate": "2026-07-06T09:00:00Z",
      "updatedAt": null
    },
    {
      "id": 1002,
      "groupId": 1001,
      "accountNumber": "77777",
      "symbol": "EREGL",
      "orderType": 1,
      "quantity": 40,
      "price": 22.20,
      "status": 1,
      "createdDate": "2026-07-06T09:00:00Z"
    }
  ]
}
```

C# örneği

```
// Hesabın OCO emir satırlarını listeler.
public async Task<string> GetOcoOrdersAsync(string brokerageCode, string accountNumber)
{
    var url = $"api/v1/oco-orders?brokerageCode={Uri.EscapeDataString(brokerageCode)}" +
        $"&accountNumber={Uri.EscapeDataString(accountNumber)}";
    var request = new HttpRequestMessage(HttpMethod.Get, url);
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

POST /api/v1/oco-orders

OCO emir grubu oluşturun

JWT gerekli / Scope: trade

1 Stop-Loss + en az 1 Take-Profit kademesinden oluşan yeni bir OCO grubu oluşturun. Tüm TP fiyatları SL fiyatından yüksek olmalıdır.

BIST fiyat kademeleri: Fiyat adımı fiyata göre değişir: 0,01-19,99 → 0,01 · 20-49,99 → 0,02 · 50-99,99 → 0,05 · 100-249,99 → 0,10 · 250-499,99 → 0,25 · 500-999,99 → 0,50. Kademeye uymayan fiyat 400 döner.

Rate limit: Emir iletimi kullanıcı + aracı kurum başına kurum kotasına tabidir (Marbaş Menkul: saatte 500 emir). Tüm limitler için Rate limiting bölümüne bakın.

Parametreler

Alan	Tip / kullanım	Açıklama
brokerageCode	body / string / zorunlu	Aracı kurum kodu.
accountNumber	body / string / zorunlu	Hesap numarası.
symbol	body / string / zorunlu	Hisse kodu (örn. "EREGL").
stopLossPrice	body / decimal / zorunlu	Zarar kes fiyatı. BIST kademesine uygun, tüm TP'lerden düşük.
stopLossQuantity	body / integer / zorunlu	Stop-Loss lot adedi.
takeProfitItems	body / array / zorunlu	Kademeler: { price, quantity }. En az 1 adet.
dueDate	body / datetime / opsiyonel	Bu tarihten sonra emirler pasife alınır.
notes	body / string / opsiyonel	Serbest not.

İstek gövdesi

```
{
  "brokerageCode": "TradeWiseYatirim",
  "accountNumber": "77777",
  "symbol": "EREGL",
  "stopLossPrice": 21.00,
  "stopLossQuantity": 100,
  "takeProfitItems": [
    { "price": 22.20, "quantity": 40 },
    { "price": 23.00, "quantity": 40 },
    { "price": 24.00, "quantity": 20 }
  ]
}
```

cURL

```
curl -X POST https://api.tradewize.com.tr/api/v1/oco-orders \
-H "Authorization: Bearer <access_token>" \
-H "Content-Type: application/json" \
-d '{
  "brokerageCode": "TradewiseYatirim",
  "accountNumber": "77777",
  "symbol": "EREGL",
  "stopLossPrice": 21.00,
  "stopLossQuantity": 100,
  "takeProfitItems": [
    { "price": 22.20, "quantity": 40 },
    { "price": 24.00, "quantity": 60 }
  ]
}'
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": {
    "id": 1042,
    "groupId": 1042,
    "accountNumber": "77777",
    "symbol": "EREGL",
    "orderType": 2,
    "quantity": 100,
    "price": 21.00,
    "status": 1,
    "createdDate": "2026-07-06T14:00:00Z"
  }
}
```

C# örneği

```
// OCO grubu oluşturur (1 SL + kademeli TP).
public record TakeProfitItem(decimal Price, int Quantity);

public async Task<string> CreateOcoOrderAsync(
    string brokerageCode, string accountNumber, string symbol,
    decimal stopLossPrice, int stopLossQuantity, IEnumerable<TakeProfitItem> takeProfitItems)
{
    var request = new HttpRequestMessage(HttpMethod.Post, "/api/v1/oco-orders");
    request.Headers.Authorization = await GetAuthHeaderAsync();
    request.Content = new StringContent(
        JsonSerializer.Serialize(new
        {
            brokerageCode, accountNumber, symbol,
            stopLossPrice, stopLossQuantity, takeProfitItems
        })),
        Encoding.UTF8, "application/json");

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

POST /api/v1/oco-orders/register-pending

Bekleyen OCO bağlantısı kaydet

JWT gerekli / Scope: trade

Bir broker emrine OCO bağlar. Emir Completed olduğunda gerçekleşme fiyatı üzerinden SL/TP fiyatları otomatik hesaplanır. SL ve her TP için yüzde (percent) VEYA fiyat (price) girin.

info: Yanıttaki data, oluşturulan bağlantının kimliğidir (setupId). SL/TP için percent ve price aynı anda verilemez; en az biri girilmelidir.

Rate limit: Emir iletimi kullanıcı + aracı kurum başına kurum kotasına tabidir (Marbaş Menkul: saatte 500 emir). Tüm limitler için Rate limiting bölümüne bakın.

Parametreler

Alan	Tip / kullanım	Açıklama
brokerageCode	body / string / zorunlu	Aracı kurum kodu.
accountNumber	body / string / zorunlu	Hesap numarası.
brokerOrderNo	body / string / zorunlu	OCO'nun bağlanacağı broker emir numarası.
symbol	body / string / zorunlu	Hisse kodu.
stopLossQuantity	body / integer / zorunlu	Stop-Loss lot adedi.
stopLossPercent	body / decimal / opsiyonel	SL yüzdesi. stopLossPrice ile birlikte verilemez; biri zorunlu.
stopLossPrice	body / decimal / opsiyonel	Sabit SL fiyatı (BIST kademesine uygun). stopLossPercent ile birlikte verilemez.
takeProfitItems	body / array / zorunlu	Kademeler: { quantity, percent price }. En az 1 adet.
dueDate	body / datetime / opsiyonel	Bu tarihten sonra satırlar pasife alınır.
notes	body / string / opsiyonel	Serbest not.

İstek gövdesi

```
{
  "brokerageCode": "TradeWiseYatirim",
  "accountNumber": "77777",
  "brokerOrderNo": "12345678",
  "symbol": "EREGL",
  "stopLossQuantity": 100,
  "stopLossPercent": 3,
  "takeProfitItems": [
    { "quantity": 40, "percent": 4.5 },
    { "quantity": 40, "price": 23.00 },
    { "quantity": 20, "percent": 8 }
  ]
}
```

cURL

```
curl -X POST https://api.tradewize.com.tr/api/v1/oco-orders/register-pending \
-H "Authorization: Bearer <access_token>" \
-H "Content-Type: application/json" \
-d '{
  "brokerageCode": "TradeWiseYatirim",
  "accountNumber": "77777",
  "brokerOrderNo": "12345678",
  "symbol": "EREGL",
  "stopLossQuantity": 100,
  "stopLossPercent": 3,
  "takeProfitItems": [
    { "quantity": 60, "percent": 4.5 },
    { "quantity": 40, "price": 23.00 }
  ]
}'
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": 5012
}
```

C# örneği

```
// Broker emrine bekleyen OCO bağlar. SL/TP için percent VEYA price girin.
public async Task<string> RegisterPendingOcoOrderAsync(object body)
{
    var request = new HttpRequestMessage(HttpMethod.Post, "/api/v1/oco-orders/register-pending");
    request.Headers.Authorization = await GetAuthHeaderAsync();
    request.Content = new StringContent(
        JsonSerializer.Serialize(body),
        Encoding.UTF8, "application/json");

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}

// Kullanım
await client.RegisterPendingOcoOrderAsync(new
{
    brokerageCode = "TradeWiseYatirim",
    accountNumber = "777777",
    brokerOrderNo = "12345678",
    symbol = "EREGL",
    stopLossQuantity = 100,
    stopLossPercent = 3,
    takeProfitItems = new object[]
    {
        new { quantity = 60, percent = 4.5 },
        new { quantity = 40, price = 23.00 }
    }
});
```

DELETE /api/v1/oco-orders/{id}

OCO grubunu iptal et

JWT gerekli / Scope: trade

Belirtilen satırın ait olduğu OCO grubundaki tüm aktif emirleri iptal eder.

Parametreler

Alan	Tip / kullanım	Açıklama
id	path / integer / zorunlu	İptal edilecek OCO satırının/grubunun kimliği.
brokerageCode	query / string / zorunlu	Aracı kurum kodu.

cURL

```
curl -X DELETE "https://api.tradewize.com.tr/api/v1/oco-orders/1042?brokerageCode=TradeWiseYatirim" \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": true
}
```

C# örneği

```
// OCO grubunu iptal eder.
public async Task<string> CancelOcoOrderAsync(int id, string brokerageCode)
{
    var url = $"/api/v1/oco-orders/{id}?brokerageCode={Uri.EscapeDataString(brokerageCode)}";
    var request = new HttpRequestMessage(HttpMethod.Delete, url);
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

Sistemler

Sistemler, bir hesaba bağlı işlem botlarının üst kümesidir. Bu uç tüm bot tiplerini tek listede döndürür (finansal performans metrikleri hariç). botType: 1 = LevelBot, 2 = TradingViewBot, 3 = StockBot, 4 = SignalBot.

POST /api/v1/systems/list

Sistemleri listele

JWT gerekli / Scope: report

Bir hesaba ait tüm sistemleri (botları) filtreleyerek döndürür.

Parametreler

Alan	Tip / kullanım	Açıklama
brokerageCode	body / string / zorunlu	Aracı kurum kodu.
accountNumber	body / string / zorunlu	Hesap numarası.
botType	body / integer[] / opsiyonel	Bot tipi filtresi (boş = tümü). 1 LevelBot, 2 TradingViewBot, 3 StockBot, 4 SignalBot.
isActive	body / integer / opsiyonel	1 = tümü, 2 = yalnızca aktif, 3 = yalnızca pasif. Varsayılan 1.
isDeleted	body / integer / opsiyonel	1 = tümü, 2 = silinmemiş, 3 = silinmiş. Varsayılan 2.
startDate	body / datetime / opsiyonel	Tarih aralığı başlangıcı (opsiyonel).
endDate	body / datetime / opsiyonel	Tarih aralığı bitişi (opsiyonel).

İstek gövdesi

```
{
  "brokerageCode": "TradeWiseYatirim",
  "accountNumber": "77777",
  "botType": [],
  "isActive": 1,
  "isDeleted": 2,
  "startDate": null,
  "endDate": null
}
```

cURL

```
curl -X POST https://api.tradewize.com.tr/api/v1/systems/list \
-H "Authorization: Bearer <access_token>" \
-H "Content-Type: application/json" \
-d '{"brokerageCode": "TradeWiseYatirim", "accountNumber": "77777", "botType": [], "isActive": 1, "isDeleted": 2}'
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": [
    {
      "id": 42,
      "name": "THYAO Level Bot",
      "description": "",
      "isActive": true,
      "realBot": true,
      "botType": 1,
      "isDeleted": false,
      "sourceType": 1,
      "symbolList": ["THYAO"],
      "accountName": "Ana Hesap",
      "accountNumber": "77777",
      "buyGap": 50,
      "sellGap": 50,
      "takeProfit": 200,
      "levelBotDetails": [
        {
          "symbol": "THYAO",
          "maxBuyPrice": 280.00,
          "minBuyPrice": 220.00,
          "maxSellPrice": 300.00,
          "minSellPrice": 230.00,
          "profit": 5,
          "isActive": true
        }
      ]
    }
  ]
}
```

C# örneği

```
// Hesaba ait tüm botları (sistemleri) listeler.
public async Task<string> GetSystemsAsync(
    string brokerageCode, string accountNumber, int[]? botType = null,
    int isActive = 1, int isDeleted = 2)
{
    var request = new HttpRequestMessage(HttpMethod.Post, "/api/v1/systems/list");
    request.Headers.Authorization = await GetAuthHeaderAsync();
    request.Content = new StringContent(
        JsonSerializer.Serialize(new
        {
            brokerageCode, accountNumber, botType = botType ?? Array.Empty<int>(),
            isActive, isDeleted
        }),
        Encoding.UTF8, "application/json");

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

Level Bot'lar

Level Bot, belirli fiyat seviyelerinde otomatik alım-satım yapan bottur. Her bot birden çok sembol için bağımsız konfigürasyon içerebilir. Fiyat limitleri konfigürasyondur; gerçek işlem fiyatları ve komisyonlar yanıtlarda paylaşılmaz.

GET /api/v1/level-bots

Level Bot'ları listele

JWT gerekli / Scope: report

Hesaba ait tüm Level Bot'ları döndürür.

Parametreler

Alan	Tip / kullanım	Açıklama
brokerageCode	query / string / zorunlu	Aracı kurum kodu.
accountNumber	query / string / zorunlu	Hesap numarası.
withPassive	query / integer / opsiyonel	Pasif botları dahil et. 1 = evet, 0 = hayır. Varsayılan 1.
withDeleted	query / integer / opsiyonel	Silinmiş botları dahil et. 1 = evet, 0 = hayır. Varsayılan 1.

cURL

```
curl -X GET "https://api.tradewize.com.tr/api/v1/level-bots?brokerageCode=TradeWiseYatirim&accountNumber=77777&withPassive=1&withDeleted=0" \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": [
    {
      "id": 42,
      "name": "THYAO-EREGL Stratejisi",
      "description": "",
      "isActive": true,
      "realBot": true,
      "botType": 1,
      "accountNumber": "77777",
      "levelBotType": 1,
      "levelType": 1,
      "buyGap": 50,
      "sellGap": 50,
      "targetProfit": 200,
      "levelBotDetails": [
        {
          "id": 501,
          "symbol": "THYAO",
          "maxBuyPrice": 280.00,
          "minBuyPrice": 220.00,
          "maxSellPrice": 300.00,
          "minSellPrice": 230.00,
          "buyQuantity": 100,
          "sellQuantity": 100,
          "profit": 5,
          "isActive": true
        }
      ]
    }
  ]
}
```

C# örneği

```
// Hesaba ait Level Bot'ları listeler.
public async Task<string> GetLevelBotsAsync(
    string brokerageCode, string accountNumber, int withPassive = 1, int withDeleted = 1)
{
    var url = $"/api/v1/level-bots?brokerageCode={Uri.EscapeDataString(brokerageCode)}" +
        $"&accountNumber={Uri.EscapeDataString(accountNumber)}" +
        $"&withPassive={withPassive}&withDeleted={withDeleted}";
    var request = new HttpRequestMessage(HttpMethod.Get, url);
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

GET /api/v1/level-bots/{id}

Level Bot getir

JWT gerekli / Scope: report

ID ile tek bir Level Bot'un tüm konfigürasyonunu döndürür.

Parametreler

Alan	Tip / kullanım	Açıklama
id	path / integer / zorunlu	Level Bot kimliği.
brokerageCode	query / string / zorunlu	Aracı kurum kodu.

cURL

```
curl -X GET "https://api.tradewize.com.tr/api/v1/level-bots/42?brokerageCode=TradeWiseYatirim" \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": {
    "id": 42,
    "name": "THYAO-EREGL Stratejisi",
    "isActive": true,
    "botType": 1,
    "accountNumber": "77777",
    "levelBotDetails": [
      { "id": 501, "symbol": "THYAO", "maxBuyPrice": 280.00, "minBuyPrice": 220.00, "profit": 5, "isActive": true }
    ]
  }
}
```

C# örneği

```
// ID ile Level Bot getirir.
public async Task<string> GetLevelBotAsync(int id, string brokerageCode)
{
    var url = $"/api/v1/level-bots/{id}?brokerageCode={Uri.EscapeDataString(brokerageCode)}";
    var request = new HttpRequestMessage(HttpMethod.Get, url);
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

POST /api/v1/level-bots

Level Bot oluřtur

JWT gerekli / Scope: manage

Yeni bir Level Bot oluřturur. levelBotDetails içinde her sembol için fiyat limitleri ve adetler tanımlanır.

Parametreler

Alan	Tip / kullanım	Açıklama
brokerageCode	body / string / zorunlu	Aracı kurum kodu.
accountNumber	body / string / zorunlu	Hesap numarası.
name	body / string / zorunlu	Bot adı.
isActive	body / boolean / zorunlu	Bot aktif mi?
sourceType	body / integer / zorunlu	1 = sembol listesi, 2 = liste, 3 = tarama.
levelBotDetails	body / array / zorunlu	Sembol konfigürasyonları: symbol, maxBuyPrice, minBuyPrice, maxSellPrice, minSellPrice, buyQuantity, sellQuantity, profit, isActive.

İstek gövdesi

```
{
  "brokerageCode": "TradeWiseYatirim",
  "accountNumber": "77777",
  "name": "THYAO Alım-Satım Botu",
  "description": "",
  "isActive": true,
  "realBot": true,
  "sourceType": 1,
  "symbolList": ["THYAO"],
  "levelBotType": 1,
  "levelType": 1,
  "buyGap": 50,
  "sellGap": 50,
  "targetProfit": 200,
  "botStartTimeString": "10:00",
  "botEndTimeString": "18:00",
  "levelBotDetails": [
    {
      "symbol": "THYAO",
      "maxBuyPrice": 280.00,
      "minBuyPrice": 220.00,
      "maxSellPrice": 300.00,
      "minSellPrice": 230.00,
      "buyQuantity": 100,
      "sellQuantity": 100,
      "profit": 5,
      "isActive": true
    }
  ]
}
```

cURL

```
curl -X POST https://api.tradewize.com.tr/api/v1/level-bots \
-H "Authorization: Bearer <access_token>" \
-H "Content-Type: application/json" \
-d @level-bot.json
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": {
    "id": 43,
    "name": "THYAO Alım-Satım Botu",
    "isActive": true,
    "accountNumber": "77777"
  }
}
```

C# örneği

```
// Yeni Level Bot oluşturur. Tam gövdeyi anonim nesne olarak geçirin.
public async Task<string> CreateLevelBotAsync(object body)
{
    var request = new HttpRequestMessage(HttpMethod.Post, "/api/v1/level-bots");
    request.Headers.Authorization = await GetAuthHeaderAsync();
    request.Content = new StringContent(
        JsonSerializer.Serialize(body),
        Encoding.UTF8, "application/json");

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

PUT /api/v1/level-bots/{id}

Level Bot güncelle

JWT gerekli / Scope: manage

Var olan Level Bot'u URL'deki kimlik üzerinden kısmi olarak günceller. Fiyat sınırları ve kâr alanlarının güncellenmesi düzeltildi; gönderilmeyen bot ve detay alanları mevcut değerlerini korur.

info: Önerilen rota PUT /api/v1/level-bots/{id} rotasıdır. Geriye uyumluluk için PUT /api/v1/level-bots desteklenmeye devam eder; legacy rotada root id gövdede zorunludur.

Kısmi güncelleme: Mevcut detay satırı id ile eşleştirilir. maxBuyPrice, minBuyPrice, maxSellPrice, minSellPrice ve profit alanları güncellenebilir; gönderilmeyen değerler 0 veya false ile ezilmez. Test ortamındaki istekler de gerçek bot ayarlarını değiştirir.

info: Root id Level Bot kimliğidir; levelBotDetails[].id ise detail satırı kimliğidir. Canonical rotada root id opsiyoneldir, ancak gönderilirse path ID ile aynı olmalıdır. Çakışma 400 döndürür.

warning: targetProfit root parent alanı, profit ise detail alanıdır. Her ikisi de tam sayı semantiğine sahiptir: 5 ve 5.0 kabul edilir, 5.5 reddedilir.

Parametreler

Alan	Tip / kullanım	Açıklama
id	path / integer / zorunlu	Güncellenecek Level Bot kimliği (bot ID).
id (body)	body / integer / opsiyonel	Opsiyonel root bot ID. Gönderilirse path ID ile aynı olmalıdır; farklıysa 400 döner.
brokerageCode	body / string / zorunlu	Aracı kurum kodu.
name	body / string / opsiyonel	Yeni bot adı.
isActive	body / boolean / opsiyonel	Bot aktif mi?
buyGap / sellGap	body / integer / opsiyonel	Bot seviyesindeki alış/satış aralıkları. Atlanırsa mevcut değerler korunur.
targetProfit	body / number / opsiyonel	Bot genelindeki hedef kâr. Root parent alanıdır ve tam sayı değerli olmalıdır.
levelBotDetails	body / array / opsiyonel	Sembol konfigürasyonları nested olarak gönderilir. Mevcut satırlar detail id ile eşleştirilir; atlanan detail alanları korunur.

İstek gövdesi

```
{
  "id": 43,
  "brokerageCode": "TradeWiseYatirim",
  "name": "THYAO Güncellenmiş Bot",
  "isActive": true,
  "buyGap": 2,
  "sellGap": 6,
  "targetProfit": 5,
  "levelBotDetails": [
    {
      "id": 501,
      "levelBotId": 43,
      "symbol": "THYAO",
      "maxBuyPrice": 290.00,
      "minBuyPrice": 210.00,
      "maxSellPrice": 310.00,
      "minSellPrice": 225.00,
      "buyQuantity": 100,
      "sellQuantity": 100,
      "profit": 5,
      "isActive": true
    }
  ]
}
```

cURL

```
curl -X PUT https://api.tradewize.com.tr/api/v1/level-bots/43 \
-H "Authorization: Bearer <access_token>" \
-H "Content-Type: application/json" \
-d @level-bot-update.json
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": {
    "id": 43,
    "name": "THYAO Güncellenmiş Bot",
    "isActive": true,
    "accountNumber": "77777",
    "sourceType": 1,
    "symbolList": ["THYAO"],
    "buyGap": 2,
    "sellGap": 6,
    "targetProfit": 5,
    "levelBotDetails": [
      {
        "id": 501,
        "levelBotId": 43,
        "symbol": "THYAO",
        "maxBuyPrice": 290.00,
        "minBuyPrice": 210.00,
        "maxSellPrice": 310.00,
        "minSellPrice": 225.00,
        "buyQuantity": 100,
        "sellQuantity": 100,
        "profit": 5,
        "isActive": true
      }
    ]
  }
}
```

C# örneği

```
// Var olan Level Bot'u canonical path ID rotasıyla kısmi günceller.
public async Task<string> UpdateLevelBotAsync(int id, object body)
{
  var request = new HttpRequestMessage(HttpMethod.Put, $"/api/v1/level-bots/{id}");
  request.Headers.Authorization = await GetAuthHeaderAsync();
  request.Content = new StringContent(
    JsonSerializer.Serialize(body),
    Encoding.UTF8, "application/json");

  var response = await _httpClient.SendAsync(request);
  response.EnsureSuccessStatusCode();
  return await response.Content.ReadAsStringAsync();
}
```

GET /api/v1/level-bots/{id}/pair-orders

Aktif çift emirler

JWT gerekli / Scope: report

Bir Level Bot'un bekleyen/işlemdeki çift emirlerini döndürür. Gerçek işlem fiyatları ve komisyonlar paylaşılmaz.

info: buyStatus / sellStatus: 1 = beklemede, 2 = emir gönderildi, 3 = tamamlandı, 4 = iptal, 5 = piyasa satış.

Parametreler

Alan	Tip / kullanım	Açıklama
id	path / integer / zorunlu	Level Bot kimliği.
brokerageCode	query / string / zorunlu	Aracı kurum kodu.

cURL

```
curl -X GET "https://api.tradewize.com.tr/api/v1/level-bots/42/pair-orders?brokerageCode=TradeWiseYatirim" \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": [
    {
      "id": 1001,
      "symbol": "THYAO",
      "buyQuantity": 100,
      "buyStatus": 3,
      "buyOrderNo": "305344701",
      "buyOrderTime": "2026-07-06T09:30:00Z",
      "buyCompleteTime": "2026-07-06T09:31:00Z",
      "sellQuantity": 100,
      "sellStatus": 1,
      "sellOrderNo": "",
      "sellOrderTime": null,
      "systemId": 42,
      "accountNumber": "77777",
      "completed": false,
      "description": "Çift emir bekleniyor"
    }
  ]
}
```

C# örneği

```
// Level Bot'un aktif çift emirlerini getirir.
public async Task<string> GetLevelBotPairOrdersAsync(int id, string brokerageCode)
{
    var url = $"/api/v1/level-bots/{id}/pair-orders?brokerageCode={Uri.EscapeDataString(brokerageCode)}";
    var request = new HttpRequestMessage(HttpMethod.Get, url);
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

GET /api/v1/level-bots/{id}/transaction-history

İşlem geçmişi

JWT gerekli / Scope: report

Bir Level Bot'un tamamlanmış işlem geçmişini sayfalı olarak döndürür. Gerçek fiyatlar, komisyon ve kâr rakamları paylaşılmaz.

Parametreler

Alan	Tip / kullanım	Açıklama
id	path / integer / zorunlu	Level Bot kimliği.
brokerageCode	query / string / zorunlu	Aracı kurum kodu.
startDate	query / datetime / zorunlu	Tarih aralığı başlangıcı (ISO 8601).
endDate	query / datetime / zorunlu	Tarih aralığı bitişi (ISO 8601).
page	query / integer / opsiyonel	Sayfa numarası. Varsayılan 1.
pageSize	query / integer / opsiyonel	Sayfa başına kayıt. Varsayılan 10.

cURL

```
curl -X GET "https://api.tradewize.com.tr/api/v1/level-bots/42/transaction-history?brokerageCode=TradeWiseYatirim&startDate=2026-07-01T00:00:00Z&endDate=2026-07-06T23:59:59Z&page=1&pageSize=20" \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": {
    "items": [
      {
        "symbol": "THYAO",
        "buyQuantity": 100,
        "buyOrderNo": "305344701",
        "buyCompleteTime": "2026-07-04T09:31:00Z",
        "sellQuantity": 100,
        "sellOrderNo": "305344702",
        "sellCompleteTime": "2026-07-04T14:20:00Z",
        "description": "Tamamlandı",
        "hourCount": 5,
        "systemId": 42
      }
    ],
    "totalCount": 125,
    "page": 1,
    "pageSize": 20
  }
}
```

C# örneği

```
// Level Bot işlem geçmişini sayfalı getirir.
public async Task<string> GetLevelBotHistoryAsync(
    int id, string brokerageCode, DateTime startDate, DateTime endDate,
    int page = 1, int pageSize = 10)
{
    var url = $"/api/v1/level-bots/{id}/transaction-history" +
        $"?brokerageCode={Uri.EscapeDataString(brokerageCode)}" +
        $"&startDate={startDate:O}&endDate={endDate:O}&page={page}&pageSize={pageSize}";
    var request = new HttpRequestMessage(HttpMethod.Get, url);
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

TradingView Bot'lar

TradingView sinyal botları, TradingView webhook sinyallerini alıp otomatik emir verir. Sinyaller, tanımlı algo adı ve periyoda göre eşleştirilir.

GET /api/v1/trading-view-bots

TradingView Bot'ları listele

JWT gerekli / Scope: report

Hesaba ait tüm TradingView Bot'ları döndürür.

Parametreler

Alan	Tip / kullanım	Açıklama
brokerageCode	query / string / zorunlu	Aracı kurum kodu.
accountNumber	query / string / zorunlu	Hesap numarası.
withPassive	query / integer / opsiyonel	Pasif botları dahil et. 1 = evet, 0 = hayır. Varsayılan 1.
withDeleted	query / integer / opsiyonel	Silinmiş botları dahil et. 1 = evet, 0 = hayır. Varsayılan 1.

cURL

```
curl -X GET "https://api.tradewize.com.tr/api/v1/trading-view-bots?brokerageCode=TradeWiseYatirim&accountNumber=77777" \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": [
    {
      "id": 10,
      "name": "THYAO TradingView",
      "isActive": true,
      "botType": 2,
      "accountNumber": "77777",
      "quantity": 100,
      "stopLoss": 3,
      "takeProfit": 5,
      "tradingViewBotDetails": [
        {
          "id": 700,
          "symbol": "THYAO",
          "maxBuyPrice": 280.00,
          "minBuyPrice": 210.00,
          "maxSellPrice": 300.00,
          "minSellPrice": 220.00,
          "quantity": 100,
          "isActive": true
        }
      ]
    }
  ]
}
```

C# örneği

```
// Hesaba ait TradingView Bot'ları listeler.
public async Task<string> GetTradingViewBotsAsync(
    string brokerageCode, string accountNumber, int withPassive = 1, int withDeleted = 1)
{
    var url = $"/api/v1/trading-view-bots?brokerageCode={Uri.EscapeDataString(brokerageCode)}" +
        $"&accountNumber={Uri.EscapeDataString(accountNumber)}" +
        $"&withPassive={withPassive}&withDeleted={withDeleted}";
    var request = new HttpRequestMessage(HttpMethod.Get, url);
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

GET /api/v1/trading-view-bots/{id}

TradingView Bot getir

JWT gerekli / Scope: report

ID ile tek bir TradingView Bot'un konfigürasyonunu döndürür.

Parametreler

Alan	Tip / kullanım	Açıklama
id	path / integer / zorunlu	TradingView Bot kimliği.
brokerageCode	query / string / zorunlu	Aracı kurum kodu.

cURL

```
curl -X GET "https://api.tradewize.com.tr/api/v1/trading-view-bots/10?brokerageCode=TradeWiseYatirim" \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": {
    "id": 10,
    "name": "THYAO TradingView",
    "isActive": true,
    "botType": 2,
    "accountNumber": "77777",
    "quantity": 100,
    "tradingViewBotDetails": [
      { "id": 700, "symbol": "THYAO", "quantity": 100, "isActive": true }
    ]
  }
}
```

C# örneği

```
// ID ile TradingView Bot getirir.
public async Task<string> GetTradingViewBotAsync(int id, string brokerageCode)
{
    var url = $"/api/v1/trading-view-bots/{id}?brokerageCode={Uri.EscapeDataString(brokerageCode)}";
    var request = new HttpRequestMessage(HttpMethod.Get, url);
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

POST /api/v1/trading-view-bots

TradingView Bot oluştur

JWT gerekli / Scope: manage

Yeni bir TradingView Bot oluşturur. sourceType 0 tüm sinyaller, 1 sembol (detay), 2 liste, 3 tarama anlamına gelir.

Parametreler

Alan	Tip / kullanım	Açıklama
brokerageCode	body / string / zorunlu	Aracı kurum kodu.
accountNumber	body / string / zorunlu	Hesap numarası.
name	body / string / zorunlu	Bot adı.
isActive	body / boolean / zorunlu	Bot aktif mi?
sourceType	body / integer / zorunlu	0 = tüm sinyaller, 1 = sembol, 2 = liste, 3 = tarama.
quantity	body / integer / opsiyonel	Sinyal başına emir adedi (lot).
tradingViewBotDetails	body / array / opsiyonel	Sembol konfigürasyonları: symbol, maxBuyPrice, minBuyPrice, maxSellPrice, minSellPrice, quantity, isActive.

İstek gövdesi

```
{
  "brokerageCode": "TradeWiseYatirim",
  "accountNumber": "77777",
  "name": "THYAO TradingView",
  "description": "",
  "isActive": true,
  "realBot": true,
  "sourceType": 1,
  "algoId": 12,
  "algoPeriodId": 3,
  "quantity": 100,
  "stopLoss": 3,
  "takeProfit": 5,
  "tradingViewBotDetails": [
    {
      "symbol": "THYAO",
      "maxBuyPrice": 280.00,
      "minBuyPrice": 210.00,
      "maxSellPrice": 300.00,
      "minSellPrice": 220.00,
      "quantity": 100,
      "isActive": true
    }
  ]
}
```

cURL

```
curl -X POST https://api.tradewize.com.tr/api/v1/trading-view-bots \
-H "Authorization: Bearer <access_token>" \
-H "Content-Type: application/json" \
-d @tv-bot.json
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": {
    "id": 11,
    "name": "THYAO TradingView",
    "isActive": true,
    "accountNumber": "77777"
  }
}
```

C# örneği

```
// Yeni TradingView Bot oluşturur.
public async Task<string> CreateTradingViewBotAsync(object body)
{
    var request = new HttpRequestMessage(HttpMethod.Post, "/api/v1/trading-view-bots");
    request.Headers.Authorization = await GetAuthHeaderAsync();
    request.Content = new StringContent(
        JsonSerializer.Serialize(body),
        Encoding.UTF8, "application/json");

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

PUT /api/v1/trading-view-bots

TradingView Bot güncelle

JWT gerekli / Scope: manage

Var olan bir TradingView Bot'u günceller. Gövde id alanını içerir.

Parametreler

Alan	Tip / kullanım	Açıklama
id	body / integer / zorunlu	Güncellenecek bot kimliği.
brokerageCode	body / string / zorunlu	Aracı kurum kodu.
name	body / string / zorunlu	Bot adı.
isActive	body / boolean / zorunlu	Bot aktif mi?
tradingViewBotDetails	body / array / opsiyonel	Güncellenmiş sembol konfigürasyonları (var olan satırlar için id ile).

İstek gövdesi

```
{
  "id": 11,
  "brokerageCode": "TradeWiseYatirim",
  "name": "THYAO TradingView v2",
  "isActive": true,
  "quantity": 150,
  "tradingViewBotDetails": [
    { "id": 700, "symbol": "THYAO", "quantity": 150, "isActive": true }
  ]
}
```

cURL

```
curl -X PUT https://api.tradewize.com.tr/api/v1/trading-view-bots \
-H "Authorization: Bearer <access_token>" \
-H "Content-Type: application/json" \
-d @tv-bot-update.json
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": true
}
```

C# örneği

```
// Var olan TradingView Bot'u günceller (gövdede id zorunlu).
public async Task<string> UpdateTradingViewBotAsync(object body)
{
    var request = new HttpRequestMessage(HttpMethod.Put, "/api/v1/trading-view-bots");
    request.Headers.Authorization = await GetAuthHeaderAsync();
    request.Content = new StringContent(
        JsonSerializer.Serialize(body),
        Encoding.UTF8, "application/json");

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

GET /api/v1/trading-view-bots/{id}/pair-orders

Çift emirler

JWT gerekli / Scope: report

Bir TradingView Bot'un çift emirlerini hesaba göre döndürür. Gerçek fiyatlar, komisyonlar ve webhook içerikleri paylaşılmaz.

info: algoName ve period, TradingView webhook sinyaliyle eşleşmelidir. buyStatus / sellStatus: 1 beklemede, 2 gönderildi, 3 tamamlandı, 4 iptal, 5 piyasa satış.

Parametreler

Alan	Tip / kullanım	Açıklama
id	path / integer / zorunlu	TradingView Bot (sistem) kimliği.
brokerageCode	query / string / zorunlu	Aracı kurum kodu.
accountNumber	query / string / zorunlu	Hesap numarası.
withPassive	query / integer / opsiyonel	Tamamlanmış çift emirleri dahil et. 1 = evet, 0 = hayır. Varsayılan 0.

cURL

```
curl -X GET "https://api.tradewize.com.tr/api/v1/trading-view-bots/10/pair-orders?brokerageCode=TradeWiseYatirim&accountNumber=77777&withPassive=0" \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": [
    {
      "id": 2001,
      "symbol": "THYAO",
      "algoName": "EMA_CROSS",
      "period": "1H",
      "systemId": 10,
      "accountNumber": "77777",
      "quantity": 100,
      "buyStatus": 3,
      "buyOrderNo": "305344801",
      "buyCompleteTime": "2026-07-06T10:00:00Z",
      "sellStatus": 1,
      "sellOrderNo": "",
      "completed": false,
      "description": "Sinyal bekleniyor"
    }
  ]
}
```

C# örneği

```
// TradingView Bot'un çift emirlerini getirir.
public async Task<string> GetTradingViewPairOrdersAsync(
    int id, string brokerageCode, string accountNumber, int withPassive = 0)
{
    var url = $"/api/v1/trading-view-bots/{id}/pair-orders" +
        $"?brokerageCode={Uri.EscapeDataString(brokerageCode)}" +
        $"&accountNumber={Uri.EscapeDataString(accountNumber)}&withPassive={withPassive}";
    var request = new HttpRequestMessage(HttpMethod.Get, url);
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

Listeler

Endeks, sektör, pazar ve kullanıcı listeleri. ListType: 1 = Endeks, 2 = Sektör, 3 = Pazar, 4 = Kullanıcı listesi. Okuma report, oluşturma/güncelleme manage kapsamı ister.

GET /api/v1/lists/indexes

Endeksleri listele

JWT gerekli / Scope: report

Tüm endeks listelerini (BIST 30, BIST 100 vb.) özet olarak döndürür.

cURL

```
curl -X GET https://api.tradewize.com.tr/api/v1/lists/indexes \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": [
    { "id": 1, "listTypeId": 1, "code": "XU030", "name": "BIST 30", "isActive": true, "symbolCount": 30 },
    { "id": 2, "listTypeId": 1, "code": "XU100", "name": "BIST 100", "isActive": true, "symbolCount": 100 }
  ]
}
```

C# örneği

```
// Endeks listelerini getirir. (sektörler/pazarlar için yol değişir)
public async Task<string> GetIndexesAsync()
{
    var request = new HttpRequestMessage(HttpMethod.Get, "/api/v1/lists/indexes");
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

GET /api/v1/lists/indexes/{code}

Endeks sembollerini getir

JWT gerekli / Scope: report

Belirli bir endeksin sembollerini döndürür (örn. code = XU030).

info: Aynı kalıp sektör (/sectors/{code}) ve pazar (/markets/{code}) için de geçerlidir.

Parametreler

Alan	Tip / kullanım	Açıklama
code	path / string / zorunlu	Endeks kodu (örn. "XU030").

cURL

```
curl -X GET https://api.tradewize.com.tr/api/v1/lists/indexes/XU030 \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": {
    "id": 1,
    "listTypeId": 1,
    "code": "XU030",
    "name": "BIST 30",
    "symbolCount": 30,
    "symbols": [
      { "symbolId": 101, "code": "THYAO", "name": "Türk Hava Yolları", "addedAt": "2026-01-01T00:00:00Z" },
      { "symbolId": 102, "code": "GARAN", "name": "Garanti Bankası", "addedAt": "2026-01-01T00:00:00Z" }
    ]
  }
}
```

C# örneği

```
// Bir endeks/sektör/pazarın sembollerini kodla getirir.
public async Task<string> GetIndexByCodeAsync(string code)
{
    var url = $"/api/v1/lists/indexes/{Uri.EscapeDataString(code)}";
    var request = new HttpRequestMessage(HttpMethod.Get, url);
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

GET /api/v1/lists/sectors

Sektörleri listele

JWT gerekli / Scope: report

Tüm sektör listelerini (Bankacılık, Sanayi vb.) döndürür. /sectors/{code} sembolleri verir.

cURL

```
curl -X GET https://api.tradewize.com.tr/api/v1/lists/sectors \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": [
    { "id": 20, "listTypeId": 2, "code": "BANK", "name": "Bankacılık", "symbolCount": 12 },
    { "id": 21, "listTypeId": 2, "code": "SANAYI", "name": "Sanayi", "symbolCount": 45 }
  ]
}
```

C# örneği

```
// Sektör listelerini getirir.
public async Task<string> GetSectorsAsync()
{
    var request = new HttpRequestMessage(HttpMethod.Get, "/api/v1/lists/sectors");
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

GET /api/v1/lists/markets

Pazarları listele

JWT gerekli / Scope: report

Tüm pazar listelerini (Yıldız Pazar, Ana Pazar vb.) döndürür. /markets/{code} sembolleri verir.

cURL

```
curl -X GET https://api.tradewize.com.tr/api/v1/lists/markets \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": [
    { "id": 30, "listTypeId": 3, "code": "YILDIZ", "name": "Yıldız Pazar", "symbolCount": 120 },
    { "id": 31, "listTypeId": 3, "code": "ANA", "name": "Ana Pazar", "symbolCount": 200 }
  ]
}
```

C# örneği

```
// Pazar listelerini getirir.
public async Task<string> GetMarketsAsync()
{
    var request = new HttpRequestMessage(HttpMethod.Get, "/api/v1/lists/markets");
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

GET /api/v1/lists/my

Kendi listelerim

JWT gerekli / Scope: report

Kullanıcının kendi oluşturduğu listeleri döndürür.

cURL

```
curl -X GET https://api.tradewize.com.tr/api/v1/lists/my \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": [
    { "id": 500, "listTypeId": 4, "code": "USR-500", "name": "Takip Listem", "symbolCount": 8 }
  ]
}
```

C# örneği

```
// Kullanıcının kendi listelerini getirir.
public async Task<string> GetMyListsAsync()
{
    var request = new HttpRequestMessage(HttpMethod.Get, "/api/v1/lists/my");
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

GET /api/v1/lists/my/{id}

Liste detayı

JWT gerekli / Scope: report

Tek bir kullanıcı listesini sembolleriyle birlikte döndürür.

Parametreler

Alan	Tip / kullanım	Açıklama
id	path / integer / zorunlu	Liste kimliği.

cURL

```
curl -X GET https://api.tradewize.com.tr/api/v1/lists/my/500 \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": {
    "id": 500,
    "listTypeId": 4,
    "code": "USR-500",
    "name": "Takip Listem",
    "symbolCount": 2,
    "symbols": [
      { "symbolId": 101, "code": "THYAO", "name": "Türk Hava Yolları", "addedAt": "2026-06-01T10:00:00Z" },
      { "symbolId": 205, "code": "ASELS", "name": "Aselsan", "addedAt": "2026-06-02T11:00:00Z" }
    ]
  }
}
```

C# örneği

```
// Kullanıcı listesini id ile getirir.
public async Task<string> GetMyListAsync(int id)
{
    var request = new HttpRequestMessage(HttpMethod.Get, $"/api/v1/lists/my/{id}");
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

POST /api/v1/lists/my

Liste oluştur

JWT gerekli / Scope: manage

Yeni bir kullanıcı listesi oluşturur. listTypeId kullanıcı listeleri için 4'tür.

Parametreler

Alan	Tip / kullanım	Açıklama
listTypeId	body / integer / zorunlu	Liste tipi. Kullanıcı listesi için 4.
name	body / string / zorunlu	Liste adı (maks. 150 karakter).
symbolList	body / string[] / opsiyonel	Başlangıç sembolleri (kod listesi).
isPublic	body / boolean / opsiyonel	Herkese açık mı? Varsayılan true.

İstek gövdesi

```
{
  "listTypeId": 4,
  "name": "Teknoloji Hisselerim",
  "isPublic": false,
  "symbolList": ["ASELS", "LOGO", "NETAS"]
}
```

cURL

```
curl -X POST https://api.tradewize.com.tr/api/v1/lists/my \
-H "Authorization: Bearer <access_token>" \
-H "Content-Type: application/json" \
-d '{"listTypeId": 4, "name": "Teknoloji Hisselerim", "symbolList": ["ASELS", "LOGO"]}'
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": { "id": 501, "code": "USR-501", "name": "Teknoloji Hisselerim", "listTypeId": 4 }
}
```

C# örneği

```
// Yeni kullanıcı listesi oluşturur.
public async Task<string> CreateMyListAsync(string name, IEnumerable<string> symbols, bool isPublic = false)
{
    var request = new HttpRequestMessage(HttpMethod.Post, "/api/v1/lists/my");
    request.Headers.Authorization = await GetAuthHeaderAsync();
    request.Content = new StringContent(
        JsonSerializer.Serialize(new { listTypeId = 4, name, isPublic, symbolList = symbols }),
        Encoding.UTF8, "application/json");

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

PUT /api/v1/lists/my/{id}

Liste güncelle

JWT gerekli / Scope: manage

Var olan bir kullanıcı listesini günceller (ad, görünürlük vb.).

Parametreler

Alan	Tip / kullanım	Açıklama
id	path / integer / zorunlu	Liste kimliği.
name	body / string / zorunlu	Yeni liste adı.
listTypeId	body / integer / zorunlu	Liste tipi (4).

İstek gövdesi

```
{
  "listTypeId": 4,
  "name": "Teknoloji Hisselerim (2026)",
  "isPublic": false
}
```

cURL

```
curl -X PUT https://api.tradewize.com.tr/api/v1/lists/my/501 \
-H "Authorization: Bearer <access_token>" \
-H "Content-Type: application/json" \
-d '{"listTypeId": 4, "name": "Teknoloji Hisselerim (2026)"}'
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": { "id": 501, "name": "Teknoloji Hisselerim (2026)" }
}
```

C# örneği

```
// Kullanıcı listesini günceller.
public async Task<string> UpdateMyListAsync(int id, object body)
{
    var request = new HttpRequestMessage(HttpMethod.Put, $"/api/v1/lists/my/{id}");
    request.Headers.Authorization = await GetAuthHeaderAsync();
    request.Content = new StringContent(
        JsonSerializer.Serialize(body), Encoding.UTF8, "application/json");

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

DELETE /api/v1/lists/my/{id}

Liste sil

JWT gerekli / Scope: manage

Bir kullanıcı listesini siler.

Parametreler

Alan	Tip / kullanım	Açıklama
id	path / integer / zorunlu	Silinecek liste kimliği.

cURL

```
curl -X DELETE https://api.tradewize.com.tr/api/v1/lists/my/501 \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": true
}
```

C# örneği

```
// Kullanıcı listesini siler.
public async Task<string> DeleteMyListAsync(int id)
{
    var request = new HttpRequestMessage(HttpMethod.Delete, $"/api/v1/lists/my/{id}");
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

POST /api/v1/lists/my/{id}/add-symbols

Listeye sembol ekle

JWT gerekli / Scope: manage

Bir kullanıcı listesine sembol ekler (sembol kimliği ile).

info: Sembol kimliğini (symbolId) uygulama arayüzünden veya mevcut liste uçlarından edinebilirsiniz.

Parametreler

Alan	Tip / kullanım	Açıklama
id	path / integer / zorunlu	Liste kimliği.
symbolId	body / integer / zorunlu	Eklenecek sembolün kimliği.

İstek gövdesi

```
{
  "symbolId": 205
}
```

cURL

```
curl -X POST https://api.tradewize.com.tr/api/v1/lists/my/500/add-symbols \
-H "Authorization: Bearer <access_token>" \
-H "Content-Type: application/json" \
-d '{"symbolId": 205}'
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": true
}
```

C# örneği

```
// Listeye sembol ekler.
public async Task<string> AddSymbolToListAsync(int listId, int symbolId)
{
    var request = new HttpRequestMessage(HttpMethod.Post, $"/api/v1/lists/my/{listId}/add-symbols");
    request.Headers.Authorization = await GetAuthHeaderAsync();
    request.Content = new StringContent(
        JsonSerializer.Serialize(new { symbolId }, Encoding.UTF8, "application/json");

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

DELETE /api/v1/lists/my/{id}/remove-symbols

Listeden sembol çıkar

JWT gerekli / Scope: manage

Bir kullanıcı listesinden sembol çıkarır.

Parametreler

Alan	Tip / kullanım	Açıklama
id	path / integer / zorunlu	Liste kimliği.
symbolId	body / integer / zorunlu	Çıkarılacak sembolün kimliği.

İstek gövdesi

```
{
  "symbolId": 205
}
```

cURL

```
curl -X DELETE https://api.tradewize.com.tr/api/v1/lists/my/500/remove-symbols \
-H "Authorization: Bearer <access_token>" \
-H "Content-Type: application/json" \
-d '{"symbolId": 205}'
```

Örnek yanıt

```
{
  "success": true,
  "message": "İşlem Başarılı",
  "data": true
}
```

C# örneği

```
// Listeden sembol çıkarır.
public async Task<string> RemoveSymbolFromListAsync(int listId, int symbolId)
{
    var request = new HttpRequestMessage(HttpMethod.Delete, $"/api/v1/lists/my/{listId}/remove-symbols");
    request.Headers.Authorization = await GetAuthHeaderAsync();
    request.Content = new StringContent(
        JsonSerializer.Serialize(new { symbolId }, Encoding.UTF8, "application/json");

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

Piyasa & Fiyat Verisi

PAY/VİOP son fiyatları, günlük OHLC, kapanmış mumlar, açık mumlar ve son tick verileri. Mevcut API key'leri kullanılabilir; mevcut report kapsamı bu veri endpoint'leri için yeterlidir, ek kapsam tanımlanması gerekmez. Örnek fiyatlar ve sözleşme kodları temsilidir.

GET /api/v1/market-data/last-price

PAY son fiyat - compact

JWT gerekli / Scope: report

PAY sembolleri için sembol-fiyat sözlüğü döndürür. Mevcut yanıt yapısı korundu; yeni alanlar compact yanıtı eklenmedi. Eski fiyatlar korunur; eksik, bulunamayan veya farklı ürüne ait semboller 0 döner.

503 düzeltmesi · sembol bazlı sonuç: Eksik/eski fiyat veya fiyat kaynağı/sembol listesi erişim sorunu nedeniyle toplu son fiyat isteğinin 503 dönmesi giderildi. Eski fiyat orijinal zaman bilgisiyle, kullanılabilir kaydı olmayan sembol 0 olarak döner. Bu değişiklik kimlik doğrulama, istek doğrulama, kota ve diğer endpoint hatalarını kaldırmaz.

Seçim kuralları: symbols veya all=true seçeneklerinden biri zorunludur. 100 farklı sembolün aşılması, bozuk sembol biçimi veya symbols ile all=true kullanılması 400 döner. Biçimi geçerli ama bulunamayan TSLA gibi bir sembol, diğer sonuçları etkilemeden 0 döner.

Fiyat zamanı: Düz fiyat yanıtı değişmedi. Fiyatla birlikte zaman bilgisi için /last-price/details kullanın; her sembol price ve timestamp alanlarıyla döner.

Tüm semboller seçimi: all=true seçiminde tüm sembol listesi oluşturulmıyorsa 200 ile boş veya kısmi sonuç dönebilir. X-TradeWise-Selection-Complete=false, seçimin tamamlanmadığını belirtir; boş sözlük doğrulanmış boş ürün listesi olarak yorumlanmamalıdır.

JWT + ürün grubu · 1 istek / saniye: Aynı JWT için ürünün compact/details ve symbols/all=true istekleri ortak sayacı kullanır. Sembol sayısı maliyeti değiştirmez. PAY ve VİOP sayaçları birbirinden ayrıdır; bu gruplarda ek saatlik kota uygulanmaz.

Parametreler

Alan	Tip / kullanım	Açıklama
symbols	query / string / opsiyonel	all=true kullanılmıyorsa zorunlu. Virgülle ayrılmış en fazla 100 farklı sembol. Boşluklar temizlenir, harfler büyütülür, tekrarlar tekilleştirilir. Sembol biçimi: [A-Z0-9_-]{1,64}.
all	query / boolean / opsiyonel	true ise ilgili ürünün tüm aktif sembollerini seçer; 100 sembol sınırı uygulanmaz. symbols ile birlikte kullanılamaz. Varsayılan false.

cURL

```
curl -X GET "https://api.tradewize.com.tr/api/v1/market-data/last-price?symbols=ASELS,EREGL,TSLA" \
-H "Authorization: Bearer <access_token>"
```

Tüm aktif PAY sembolleri

```
curl -X GET "https://api.tradewize.com.tr/api/v1/market-data/last-price?all=true" \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "ASELS": 370.5,
  "EREGL": 28.12,
  "TSLA": 0
}
```

C# örneği

```
// PAY compact son fiyat. symbols ile all=true mutually exclusive'dir.
public async Task<string> GetPayLastPriceAsync(string? symbols = null, bool all = false)
{
    if (all == !string.IsNullOrEmpty(symbols))
        throw new ArgumentException("symbols veya all=true seçeneklerinden tam biri gerekir.");

    var url = all
        ? "/api/v1/market-data/last-price?all=true"
        : $"/api/v1/market-data/last-price?symbols={Uri.EscapeDataString(symbols!)}";
    var request = new HttpRequestMessage(HttpMethod.Get, url);
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

GET /api/v1/market-data/last-price/details

PAY son fiyat - details

JWT gerekli / Scope: report

Her sembol için yalnız price ve timestamp döndürür. timestamp, fiyat kaydının Unix milisaniye zamanıdır; istek zamanı veya gecikme süresi değildir. Düz fiyat endpoint'ini ayrıca çağırmak gerekmez.

price ve timestamp: Yanıt doğrudan sembol sözlüğüdür; her kayıta yalnız price (number) ve timestamp (integer veya null) bulunur. timestamp, 1 Ocak 1970 başlangıcına göre milisaniyedir. Eski fiyatın orijinal zamanı korunur; kayıt yeniden yüklendiğinde veya sorgulandığında güncellenmez.

Eksik kayıt: Fiyat bulunamayan veya farklı ürüne ait sembol price=0, timestamp=null döner; diğer semboller etkilenmez. Fiyat var ama olay zamanı bilinmiyorsa fiyat korunur, timestamp=null olur.

Tüm semboller: symbols en fazla 100 farklı sembol kabul eder; tamamı için all=true kullanılır. Sembol listesi henüz hazır değilse boş sözlük dönebilir. Bu durumda X-TradeWise-Selection-Complete başlığı false olur.

JWT + ürün grubu · 1 istek / saniye: Aynı JWT için ürünün compact/details ve symbols/all=true istekleri ortak sayacı kullanır. Sembol sayısı maliyeti değiştirmez. PAY ve VIOP sayaçları birbirinden ayrıdır; bu gruplarda ek saatlik kota uygulanmaz.

Parametreler

Alan	Tip / kullanım	Açıklama
symbols	query / string / opsiyonel	all=true kullanılmıyorsa zorunlu. Virgülle ayrılmış en fazla 100 farklı sembol. Boşluklar temizlenir, harfler büyütülür, tekrarlar tekilleştirilir. Sembol biçimi: [A-Z0-9_-]{1,64}.
all	query / boolean / opsiyonel	true ise ilgili ürünün tüm aktif sembollerini seçer; 100 sembol sınırı uygulanmaz. symbols ile birlikte kullanılamaz. Varsayılan false.

cURL

```
curl -X GET "https://api.tradewize.com.tr/api/v1/market-data/last-price/details?symbols=ASELS,TSLA" \
-H "Authorization: Bearer <access_token>"
```

200 OK

```
{
  "ASELS": {
    "price": 370.5,
    "timestamp": 1788515700000
  },
  "TSLA": {
    "price": 0,
    "timestamp": null
  }
}
```

C# örneği

```
// PAY details - fiyat ve Unix milisaniye zamanını okur.
public async Task<string> GetPayLastPriceDetailsAsync(string symbols)
{
    var url = $"/api/v1/market-data/last-price/details?symbols={Uri.EscapeDataString(symbols)}";
    var request = new HttpRequestMessage(HttpMethod.Get, url);
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

GET /api/v1/market-data/viop/last-price

VIOP son fiyat - compact

JWT gerekli / Scope: report

VIOP sözleşmeleri için sembol-fiyat sözlüğü döndürür. PAY fiyatları bu endpoint'ten alınmaz. 100 sembol sınırı, all=true ve eksik/eski fiyat davranışı PAY son fiyat endpoint'iyle aynıdır.

Ürün ayrımı: PAY sembolleri /last-price, VIOP sözleşmeleri /viop/last-price üzerinden sorgulanır. Yanlış ürün seçimi ilgili sembol için 0 döner; details yanıtında price=0 ve timestamp=null olur.

503 düzeltmesi · sembol bazlı sonuç: Eksik/eski fiyat veya fiyat kaynağı/sembol listesi erişim sorunu nedeniyle toplu son fiyat isteğinin 503 dönmesi giderildi. Eski fiyat orijinal zaman bilgisiyle, kullanılabilir kaydı olmayan sembol 0 olarak döner. Bu değişiklik kimlik doğrulama, istek doğrulama, kota ve diğer endpoint hatalarını kaldırır.

JWT + ürün grubu · 1 istek / saniye: Aynı JWT için ürünün compact/details ve symbols/all=true istekleri ortak sayacı kullanır. Sembol sayısı maliyeti değiştirmez. PAY ve VIOP sayaçları birbirinden ayrıdır; bu gruplarda ek saatlik kota uygulanmaz.

Parametreler

Alan	Tip / kullanım	Açıklama
symbols	query / string / opsiyonel	all=true kullanılmıyorsa zorunlu. Virgülle ayrılmış en fazla 100 farklı sembol. Boşluklar temizlenir, harfler büyütülür, tekrarlar tekilleştirilir. Sembol biçimi: [A-Z0-9_-]{1,64}.
all	query / boolean / opsiyonel	true ise ilgili ürünün tüm aktif sembollerini seçer; 100 sembol sınırı uygulanmaz. symbols ile birlikte kullanılamaz. Varsayılan false.

cURL

```
curl -X GET "https://api.tradewize.com.tr/api/v1/market-data/viop/last-price?symbols=F_ASELS0926" \
-H "Authorization: Bearer <access_token>"
```

```
# Tüm aktif VIOP sözleşmeleri
curl -X GET "https://api.tradewize.com.tr/api/v1/market-data/viop/last-price?all=true" \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "F_ASELS0926": 380.25
}
```

GET /api/v1/market-data/viop/last-price/details

VIOP son fiyat - details

JWT gerekli / Scope: report

Her sembol için yalnız price ve timestamp döndürür. timestamp, fiyat kaydının Unix milisaniye zamanıdır; istek zamanı veya gecikme süresi değildir. Düz fiyat endpoint'ini ayrıca çağırmak gerekmez.

price ve timestamp: Yanıt doğrudan sembol sözlüğüdür; her kayıta yalnız price (number) ve timestamp (integer veya null) bulunur. timestamp, 1 Ocak 1970 başlangıcına göre milisaniyedir. Eski fiyatın orijinal zamanı korunur; kayıt yeniden yüklendiğinde veya sorgulandığında güncellenmez.

Eksik kayıt: Fiyat bulunamayan veya farklı ürüne ait sembol price=0, timestamp=null döner; diğer semboller etkilenmez. Fiyat var ama olay zamanı bilinmiyorsa fiyat korunur, timestamp=null olur.

Tüm semboller: symbols en fazla 100 farklı sembol kabul eder; tamamı için all=true kullanılır. Sembol listesi henüz hazır değilse boş sözlük dönebilir. Bu durumda X-TradeWise-Selection-Complete başlığı false olur.

JWT + ürün grubu · 1 istek / saniye: Aynı JWT için ürünün compact/details ve symbols/all=true istekleri ortak sayacı kullanır. Sembol sayısı maliyeti değiştirmez. PAY ve VIOP sayaçları birbirinden ayrıdır; bu gruplarda ek saatlik kota uygulanmaz.

Parametreler

Alan	Tip / kullanım	Açıklama
symbols	query / string / opsiyonel	all=true kullanılmıyorsa zorunlu. Virgülle ayrılmış en fazla 100 farklı sembol. Boşluklar temizlenir, harfler büyütülür, tekrarlar tekilleştirilir. Sembol biçimi: [A-Z0-9_-]{1,64}.
all	query / boolean / opsiyonel	true ise ilgili ürünün tüm aktif sembollerini seçer; 100 sembol sınırı uygulanmaz. symbols ile birlikte kullanılamaz. Varsayılan false.

cURL

```
curl -X GET "https://api.tradewize.com.tr/api/v1/market-data/viop/last-price/details?symbols=F_ASELS0926" \
-H "Authorization: Bearer <access_token>"
```

```
curl -X GET "https://api.tradewize.com.tr/api/v1/market-data/viop/last-price/details?all=true" \
-H "Authorization: Bearer <access_token>"
```

200 OK

```
{
  "F_ASELS0926": {
    "price": 380.25,
    "timestamp": 1788516000000
  }
}
```

GET /api/v1/market-data/daily-ohlc

Günlük OHLC

JWT gerekli / Scope: report

PAY/VIOP sembolleri için günlük açılış, en yüksek, en düşük, kapanış ve önceki kapanış değerlerini sembol bazlı sözlük olarak döndürür. Açık seansta oluşmakta olan, kapanış sonrası kesinleşmiş günlük bar kullanılır.

Alanlar: lastClose önceki kapanıştır. lastUpdatedUtc barın güncellenme zamanı, tradingDate işlem günüdür. isFinal=false oluşmakta olan, true kesinleşmiş barı belirtir.

Eksik veri: Verisi olmayan sembol yanıtta bulunmayabilir; sıfır değerli OHLC kaydı üretilmez. Hiç kullanılabılır veri olmaması veya geçici erişim hatası 503 döndürebilir. Son fiyat endpoint'lerindeki sembol bazlı 0 davranışı bu endpoint için geçerli değildir.

JWT + endpoint · 1 istek / 5 saniye: Aynı JWT için iki kabul edilen istek arasında en az 5.000 ms gerekir. En fazla 250 sembol tek istekte sorgulanır; sembol sayısı veya sembol değişikliği sayacı etkilemez. Sayaç live-bar, recent-ticks ve fiyat gruplarından ayırdır.

Parametreler

Alan	Tip / kullanım	Açıklama
symbols	query / string / zorunlu	Virgülle ayrılmış en fazla 250 farklı PAY/VİOP sembolü. all=true desteklenmez.

cURL

```
curl -X GET "https://api.tradewize.com.tr/api/v1/market-data/daily-ohlc?symbols=ASELS,F_ASELS0926" \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "ASELS": {
    "open": 375,
    "high": 383,
    "low": 373.5,
    "close": 380.25,
    "lastClose": 372,
    "tradingDate": "2026-09-04",
    "lastUpdatedUtc": "2026-09-04T10:00:00Z",
    "isFinal": false
  },
  "F_ASELS0926": {
    "open": 376,
    "high": 383.5,
    "low": 374,
    "close": 380.5,
    "lastClose": 373,
    "tradingDate": "2026-09-04",
    "lastUpdatedUtc": "2026-09-04T10:00:00Z",
    "isFinal": false
  }
}
```

C# örneği

```
// Günlük OHLC snapshot (tek veya çoklu sembol).
public async Task<string> GetDailyOhlcAsync(string symbols)
{
    var url = $"/api/v1/market-data/daily-ohlc?symbols={Uri.EscapeDataString(symbols)}";
    var request = new HttpRequestMessage(HttpMethod.Get, url);
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

GET /api/v1/market-data/bars

Kapanmış tarihsel barlar

JWT gerekli / Scope: report

Tek PAY veya VİOP sembolünün kapanmış OHLC barlarını getirir. Gün içi periyotlarda tradingDate, UTC aralığı veya countBack; günlük, haftalık ve aylık periyotlarda tarih aralığı, UTC aralığı veya countBack kullanılır. Açık mumlar bu yanıtın parçası değildir.

15m isteği neden 400 döner?: 15m gün içi bir periyottur. interval=15m ile fromDate/toDate kullanılamaz; tradingDate, fromUtc/toUtc veya countBack seçeneklerinden birini kullanın. fromDate/toDate yalnız 1D, 1W ve 1M periyotları içindir.

Periyot ve tarih seçimi: 1m, 3m, 5m, 15m, 30m, 60m ve 240m için: tek gün gerekiyorsa tradingDate, kesin saat aralığı gerekiyorsa fromUtc/toUtc, son N mum gerekiyorsa countBack kullanın. 1D, 1W ve 1M için tarih aralığında fromDate/toDate kullanabilirsiniz.

Sayfalama ve eksik geçmiş: Sayfa içindeki sıralama eskiden yeniye doğrudur. hasMore=true ise nextCursor, aynı sorguya cursor olarak eklenir. isPartial=true eksik geçmişi belirtir; doğrulanmış boş sonuç anlamına gelmez. noData ayrı değerlendirilmelidir.

Sadeleştirilmiş bar yanıtı: Müşteri yanıtında tickCount, barId, interval, revision, isFinal ve calendarVersion alanları yer almaz. Sorgu periyodu request içindeki interval parametresiyle belirlenmeye devam eder.

Kapanmış ve açık mum: Yanıt yalnız kesinleşmiş/kapanmış barları içerir. Oluşmakta olan mum için live-bar, son fiyat olayları için recent-ticks kullanılır. VİOP tradingDate değeri sözleşmenin işlem günüdür.

Mevcut tarihsel veri kotaları korundu: Yeni saniyelik limit eklenmedi. API key başına mevcut 120 ağırlık birimi/dakika ve kullanıcı başına 5.000 istek/saat kotaları sürer. İstek ağırlığı $1 + \text{ceil}(\text{tahmini bar sayısı} / 500)$ olarak hesaplanır. X-RateLimit-Limit, Remaining, Reset ve Request-Weight başlıklarını izleyin; 429 sonrasında Retry-After süresini bekleyin.

Hata durumları: Geçersiz parametre veya cursor 400, kota aşımı 429; geçici veri erişim veya zaman aşımı hataları 502/503/504 döndürebilir. 503 düzeltmesi yalnız son fiyat endpoint'lerindeki eksik/eski fiyat ve sembol listesi erişim davranışını kapsar.

Parametreler

Alan	Tip / kullanım	Açıklama
symbol	query / string / zorunlu	Tek PAY veya VİOP sembolü; virgülle çoklu sembol kabul edilmez. Örnek: ASELS veya F_ASELS0926.
interval	query / string / zorunlu	Gün içi: 1m, 3m, 5m, 15m, 30m, 60m, 240m. Takvim periyotları: 1D, 1W, 1M. 1m dakika, 1M ay periyodudur.
countBack	query / integer / opsiyonel	Tüm periyotlarda en güncel kapanmış barları almak için kullanılır. Örnek: interval=15m&countBack=100. limit ile birlikte verilirse değerleri aynı olmalıdır.
tradingDate	query / date / opsiyonel	Yalnız gün içi periyotlar için tek işlem günü (YYYY-MM-DD). Örnek: interval=15m&tradingDate=2026-09-04. fromDate/toDate veya fromUtc/toUtc ile birlikte kullanılamaz; 1D/1W/1M ile kullanılırsa 400 döner.
fromDate	query / date / opsiyonel	Yalnız 1D/1W/1M için dahil başlangıç tarihi. 1m-240m gün içi periyotlarla kullanılırsa 400 döner. tradingDate veya UTC aralığıyla birlikte gönderilemez.
toDate	query / date / opsiyonel	Yalnız 1D/1W/1M için hariç bitiş tarihi. Başlangıçtan sonra olmalıdır; 1m-240m gün içi periyotlarla kullanılırsa 400 döner.
fromUtc	query / datetime / opsiyonel	Tüm periyotlarda kullanılabilen alternatif dahil başlangıç zamanı. 15 dakikalık belirli bir saat aralığı için interval=15m ile fromUtc/toUtc gönderin. Tarih seçicileriyle birlikte kullanılamaz.
toUtc	query / datetime / opsiyonel	Alternatif hariç bitiş zamanı. Başlangıçtan sonra olmalıdır; fromUtc ile birlikte kullanılması önerilir.
limit	query / integer / opsiyonel	Sayfa büyüklüğü: en fazla 1.000 bar; 1M için en fazla 300 bar. countBack ile birlikte gönderilirse aynı değerde olmalıdır.
cursor	query / string / opsiyonel	Önceki yanıtta nextCursor. Aynı sembol, periyot ve aralık korunarak sonraki (daha eski) sayfa istenir.
includeOpenBar	query / boolean / opsiyonel	Yalnız false desteklenir. true isteği 400 döner; açık mum için live-bar kullanılır.

cURL

En güncel 100 adet kapanmış 15 dakikalık mum

```
curl -X GET "https://api.tradewize.com.tr/api/v1/market-data/bars?symbol=ASELS&interval=15m&countBack=100" \
-H "Authorization: Bearer <access_token>"
```

Belirli bir işlem gününün 15 dakikalık mumları

```
curl -X GET
"https://api.tradewize.com.tr/api/v1/market-data/bars?symbol=ASELS&interval=15m&tradingDate=2026-09-04&limit=1000" \
-H "Authorization: Bearer <access_token>"
```

Belirli bir UTC zaman aralığının 15 dakikalık mumları

```
curl -X GET "https://api.tradewize.com.tr/api/v1/market-data/bars?symbol=ASELS&interval=15m&fromUtc=2026-09-04T07:00:00Z
&toUtc=2026-09-04T15:10:00Z" \
-H "Authorization: Bearer <access_token>"
```

Günlük mumlarda tarih aralığı: başlangıç dahil, bitiş hariç

```
curl -X GET
"https://api.tradewize.com.tr/api/v1/market-data/bars?symbol=ASELS&interval=1D&fromDate=2026-08-01&toDate=2026-09-01" \
-H "Authorization: Bearer <access_token>"
```

200 OK · kısaltılmış yanıt (tek bar gösteriliyor)

```
{
  "symbol": "ASELS",
  "bars": [
    {
      "timeUtc": "2026-09-04T07:00:00Z",
      "closeTimeUtc": "2026-09-04T07:05:00Z",
      "open": 380,
      "high": 381,
      "low": 379.75,
      "close": 380.25
    }
  ],
  "noData": false,
  "hasMore": false,
  "nextCursor": null,
  "isPartial": false
}
```

C# örneği

```
// Kapanmış barlar: countBack, tarih/zaman aralığı ve cursor.
public async Task<string> GetHistoricalBarsAsync(
    string symbol,
    string interval,
    DateOnly? tradingDate = null,
    DateOnly? fromDate = null,
    DateOnly? toDate = null,
    int? limit = null,
    string? cursor = null,
    int? countBack = null,
    DateTimeOffset? fromUtc = null,
    DateTimeOffset? toUtc = null)
{
    var url =
        $"/api/v1/market-data/bars?symbol={Uri.EscapeDataString(symbol)}&interval={Uri.EscapeDataString(interval)}";
    if (tradingDate is not null) url += $"&tradingDate={tradingDate:yyyy-MM-dd}";
    if (fromDate is not null) url += $"&fromDate={fromDate:yyyy-MM-dd}";
    if (toDate is not null) url += $"&toDate={toDate:yyyy-MM-dd}";
    if (limit is not null) url += $"&limit={limit}";
    if (!string.IsNullOrEmpty(cursor)) url += $"&cursor={Uri.EscapeDataString(cursor)}";
    if (countBack is not null) url += $"&countBack={countBack}";
    if (fromUtc is not null) url += $"&fromUtc={Uri.EscapeDataString(fromUtc.Value.ToString("O"))}";
    if (toUtc is not null) url += $"&toUtc={Uri.EscapeDataString(toUtc.Value.ToString("O"))}";

    var request = new HttpRequestMessage(HttpMethod.Get, url);
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

GET /api/v1/market-data/live-bar

Açık mum - live-bar

JWT gerekli / Scope: report

PAY/VİOP sembollerinde seçilen periyodun oluşmakta olan OHLC barını döndürür. Kapanmış mumlar bars endpoint'inden alınır; grafik üzerinde açık mum sembol, periyot ve açılış zamanıyla güncellenir.

Alanlar ve eksik kayıt: tickCount tick sayısıdır. Aktif mumu olmayan sembol bars listesinde bulunmayabilir. isClosed=false oluşmakta olan barı belirtir. Geçici veri erişim hatası 503 döndürebilir.

JWT + endpoint · 1 istek / 5 saniye: Aynı JWT için kabul edilen istekler arasında en az 5.000 ms gerekir. Sayaç günlük OHLC ve diğer endpoint gruplarından ayrıdır. Sembol veya periyot değiştirmek limiti sıfırlamaz.

Parametreler

Alan	Tip / kullanım	Açıklama
symbols	query / string / zorunlu	Virgülle ayrılmış en fazla 250 farklı PAY/VİOP sembolü. all=true desteklenmez.
period	query / string / zorunlu	1m, 3m, 5m, 15m, 30m, 60m, 240m veya 1D.

cURL

```
curl -X GET "https://api.tradewize.com.tr/api/v1/market-data/live-bar?symbols=ASELS,F_ASELS0926&period=5m" \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "success": true,
  "period": "5m",
  "serverTimeUtc": "2026-09-04T07:03:00Z",
  "count": 1,
  "bars": [
    {
      "symbol": "ASELS",
      "period": "5m",
      "openTimeUnix": 1788505200,
      "openTimeUtc": "2026-09-04T07:00:00Z",
      "open": 380,
      "high": 381,
      "low": 379.75,
      "close": 380.25,
      "tickCount": 20,
      "firstTickTimeUtc": "2026-09-04T07:00:00.140Z",
      "lastTickTimeUtc": "2026-09-04T07:02:58Z",
      "isClosed": false
    }
  ]
}
```

C# örneği

```
// Anlık (kapanmamış) canlı bar - içinde bulunulan dakika dahil.
public async Task<string> GetLiveBarAsync(string symbols, string period)
{
    var url =
        $"/api/v1/market-data/live-bar?symbols={Uri.EscapeDataString(symbols)}&period={Uri.EscapeDataString(period)}";
    var request = new HttpRequestMessage(HttpMethod.Get, url);
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

GET /api/v1/market-data/recent-ticks

Son tick'ler - recent-ticks

JWT gerekli / Scope: report

PAY/VİOP sembollerinin son 1-5 saniyedeki fiyat olaylarını eskiden yeniye sıralı döndürür. Kısa aralıklı sorgulama için tasarlanmıştır; tarihsel tick arşivi değildir.

Tekilleştirme: Çakışan sorgu pencerelerindeki tick'ler eventId üzerinden tekilleştirilmelidir. Üç saniyede bir son beş saniyeyi sorgulamak aralıkların örtüşmesini sağlar; aynı olayın iki defa işlenmesini önlemek istemcinin sorumluluğundadır.

JWT + endpoint · 1 istek / 3 saniye: Aynı JWT için kabul edilen istekler arasında en az 3.000 ms gerekir. Sembol sayısı, sembol listesi ve seconds değişikliği sayacı sınırlamaz. Geçici veri erişim hatası 503 döndürebilir.

Parametreler

Alan	Tip / kullanım	Açıklama
symbols	query / string / zorunlu	Virgülle ayrılmış en fazla 250 farklı PAY/VİOP sembolü. all=true desteklenmez.
seconds	query / integer / opsiyonel	Geriye bakış penceresi: 1-5 saniye; varsayılan 5. İzin verilen aralık dışı değerler 400 döner.

cURL

```
curl -X GET "https://api.tradewize.com.tr/api/v1/market-data/recent-ticks?symbols=ASELS&seconds=5" \
-H "Authorization: Bearer <access_token>"
```

Örnek yanıt

```
{
  "success": true,
  "windowSeconds": 5,
  "serverTimeUtc": "2026-09-04T07:00:30Z",
  "fromMs": 1788505225000,
  "count": 1,
  "ticks": [
    {
      "eventId": "example-tick-1",
      "symbol": "ASELS",
      "price": 380.25,
      "timestampMs": 1788505228000,
      "marketType": "pay"
    }
  ]
}
```

C# örneği

```
// Son X saniyedeki ham tick'ler (X en fazla 5).
public async Task<string> GetRecentTicksAsync(string symbols, int seconds = 5)
{
    var url = $"/api/v1/market-data/recent-ticks?symbols={Uri.EscapeDataString(symbols)}&seconds={seconds}";
    var request = new HttpRequestMessage(HttpMethod.Get, url);
    request.Headers.Authorization = await GetAuthHeaderAsync();

    var response = await _httpClient.SendAsync(request);
    response.EnsureSuccessStatusCode();
    return await response.Content.ReadAsStringAsync();
}
```

Gerçek Zamanlı (SignalR / WebSocket)

Emir değişiklikleri ve portföy güncellemeleri tek SignalR hub üzerinden, iki ayrı abonelik metoduyla yayınlanır. Hub adresi `wss://api.tradewize.com.tr/hubs/positions` şeklindedir; resmi SignalR istemcisi negotiate ve WebSocket yükseltmesini otomatik yönetir. Bağlantıda JWT access token gönderilmeli, abonelikler Heartbeat() çağrısıyla canlı tutulmalıdır.

HUB /hubs/positions

SubscribeToPositions - emir değişikliklerine abone ol

JWT gerekli

Belirtilen aracı kurum hesabının emir değişiklikleri akışına abone olur ve bool sonuç döndürür. Metot adı geriye dönük uyumluluk nedeniyle SubscribeToPositions olarak kalmıştır; bu abonelik portföy değil, ReceiveOrderChanges eventindeki emir değişikliklerini yayınlar.

Gerekli yetki: JWT zorunludur ve API anahtarında read:orders kapsamı bulunmalıdır. Yalnızca JWT sahibine ait hesaplara abone olunabilir.

Metot adı ve event eşleşmesi: SubscribeToPositions çağrısından sonra ReceiveOrderChanges dinlenmelidir. Eski ReceivePositions eventi kullanılmaz.

Abonelik yaşam döngüsü: Aboneliği UnsubscribeFromPositions ile kapatın. Bağlantı kesildiğinde sunucu abonelik kaydını temizler; uzun yaşayan bağlantılarda Heartbeat() çağırın.

Parametreler

Alan	Tip / kullanım	Açıklama
brokerageCode	argument / string / zorunlu	Aracı kurum kodu (örn. MarbasMenkul).
accountNumber	argument / string / zorunlu	Abone olunacak ve JWT sahibine ait olması gereken hesap numarası.

Bağlantı ve abonelik / JavaScript

```
import * as signalR from '@microsoft/signalr';

const connection = new signalR.HubConnectionBuilder()
  .withUrl('https://api.tradewize.com.tr/hubs/positions', {
    accessTokenFactory: () => accessToken,
  })
  .withAutomaticReconnect()
  .build();

connection.on('ReceiveOrderChanges', (changes) => {
  console.log('Emir değişiklikleri', changes);
});
connection.on('Error', (message) => console.error(message));

await connection.start();
const subscribed = await connection.invoke(
  'SubscribeToPositions',
  'MarbasMenkul',
  '77777',
);

// Aktif aboneliklerin TTL süresini yenileyin.
setInterval(() => connection.invoke('Heartbeat'), 60_000);

// Aboneliği kapatmak için:
// await connection.invoke('UnsubscribeFromPositions', 'MarbasMenkul', '77777');
```

Event · ReceiveOrderChanges

```
[
  {
    "orderId": "305344701",
    "symbol": "THYAO",
    "side": "BUY",
    "quantity": 100,
    "remainingQuantity": 0,
    "visibleQuantity": 100,
    "orderType": 2,
    "status": "Completed",
    "statusDescription": "Gerçekleşti",
    "accountNumber": "77777",
    "changeType": "StatusChanged",
    "previousStatus": "Pending",
    "changedAt": "2026-08-26T10:35:22Z"
  }
]
```

HUB /hubs/positions

SubscribeToPortfolio - portföy güncellemelerine abone ol

JWT gerekli

Belirtilen aracı kurum hesabının portföy akışına abone olur ve bool sonuç döndürür. Başarılı abonelikten sonra güncel portföy nesneleri ReceivePortfolio eventiyle yayınlanır.

Gerekli yetki: JWT zorunludur ve API anahtarında read:positions kapsamı bulunmalıdır. Yalnızca JWT sahibine ait hesaplara abone olunabilir.

Abonelik yaşam döngüsü: Aboneliği UnsubscribeFromPortfolio ile kapatın. Bağlantı kesildiğinde sunucu abonelik kaydını temizler; uzun yaşayan bağlantılarda Heartbeat() çağırın.

Parametreler

Alan	Tip / kullanım	Açıklama
brokerageCode	argument / string / zorunlu	Aracı kurum kodu (örn. MarbasMenkul).
accountNumber	argument / string / zorunlu	Abone olunacak ve JWT sahibine ait olması gereken hesap numarası.

Bağlantı ve abonelik / JavaScript

```
import * as signalR from '@microsoft/signalr';

const connection = new signalR.HubConnectionBuilder()
  .withUrl('https://api.tradewize.com.tr/hubs/positions', {
    accessTokenFactory: () => accessToken,
  })
  .withAutomaticReconnect()
  .build();

connection.on('ReceivePortfolio', (portfolio) => {
  console.log('Portföy güncellemesi', portfolio);
});
connection.on('Error', (message) => console.error(message));

await connection.start();
const subscribed = await connection.invoke(
  'SubscribeToPortfolio',
  'MarbasMenkul',
  '77777',
);

// Aktif aboneliklerin TTL süresini yenileyin.
setInterval(() => connection.invoke('Heartbeat'), 60_000);

// Aboneliği kapatmak için:
// await connection.invoke('UnsubscribeFromPortfolio', 'MarbasMenkul', '77777');
```

Event · ReceivePortfolio

```
{
  "accountNumber": "77777",
  "dateTime": "2026-08-26T10:35:22Z",
  "accountBalance": 152340.50,
  "cashAmount": 12500.00,
  "t1NetCashAmount": 12500.00,
  "t2NetCashAmount": 12500.00,
  "overall": 152340.50,
  "stockOverall": 139840.50,
  "totalProfit": 8420.75,
  "positions": [
    {
      "symbol": "THYAO",
      "lot": 100,
      "sellable": 100,
      "profitRate": 6.02
    }
  ]
}
```