

BTC Trader

WebSocket Feed

Integration Manual

Sunucu Version	2.3.0
Doküman Tarih	10 Mayıs 2019
Hazırlayan	Mehmet HELVACIKÖYLÜ
2.1.0	<ul style="list-style-type: none">- Kullanıcı bakiye güncelleme modeli- Sistem mesaj modeli- Kullanıcı oturum sonlandırma modeli- Cache temizleme ve sayfayı yenileme modeli- Emir ekle, sil, güncelle modellerinin detayları- Doğrudan HTTP EndPoint'leri ile iletişim
2.2.0	<ul style="list-style-type: none">- Decimal değerler number'dan string'e çevrildi
2.3.0	<ul style="list-style-type: none">- User-Friendly Ticker HTML sayfası eklendi- Orderbook değişen emirler kanalı eklendi- Kanal, olay ve model tablosu eklendi
2.4.0	<ul style="list-style-type: none">- Tradeview eklendi

WebSocket Sunucu Bağlantısı

BTC Trader WebSocket test sunucu bilgileri aşağıdaki gibidir:

wss://ws-feed-sandbox.btctrader.com

Bağlantı SSL protokolü ile 443. porttan yapılmaktadır.

Sunucuya bağlantı kurulduktan sonraki yaşam döngüsüyle ilgili genel olarak bilinmesi gerekenleri aşağıdaki gibi sıralayabiliriz.

- Bağlantı kurulduktan sonra sunucu, istemciye doğrudan mesaj göndermez.
- İlk mesajı kullanıcı gönderir. Bu genellikle üye girişi mesajı olur.
- İstemci, üye girişi, abonelik gibi mesajlar gönderdikten sonra sunucu bu mesajlara Result model ile cevap dönecektir. Bilgi ve kontrol amaçlı olarak bu gelen veriler kullanılabilir.
- Kullanıcı girişi yapıldıktan sonra, artık sunucu tarafından bu kullanıcıya özel olan türü U ile belirtilmiş mesajlar gelebilir. İstemci bu mesajları dinlemeli, gelen mesajlara göre işlemlerini yapmalıdır.
- İstemci belirli bir olayı takip etmek istiyorsa bu olaya abone olmalıdır. Olaylar, kanalların içinde bulunur. Bir olaya abone olmak için kanal adı ve olay adı belirtilmelidir.
- Bir olaya abone olunduktan sonra bu olay ile ilgili mesajlar sürekli olarak sunucudan istemciye gönderilir.
- İstemci bu olaya ait mesajları almak istemiyorsa bu olayın aboneliğinden çıkma mesajı gönderir. Örneğin uygulamada BTCTRY pair seçili iken BTCTRY olaylarına abone olunur. Daha sonra kullanıcı ETHTRY pair'ini seçtiğinde BTCTRY aboneliklerinden çıkılmalı, ETHTRY aboneliklerine girilmelidir.
- Olaya abonelikte limit yoktur. Tüm olaylara aynı anda abone olunabilir. Ancak bu istemcinin trafiği yoğun bir şekilde kullanabilir. Kısıtlı veya ücretli bağlantılarda tüm olaylara abonelik önerilmez.

Sunucu bağlantısı ve kullanıcı giriş işlemleri ile ilgili genel olarak bilinmesi gerekenleri aşağıdaki gibi sıralayabiliriz:

- İstemci, misafir değilse kullanıcı girişi paketi gönderir.
- Misafir kullanıcılar için de giriş paketi (misafir olarak girildiğini belirtmek için) gönderilebilir. Böyle bir durumda kullanıcı id değerine 0, kullanıcı adına "guest", token değerine ise null veri girilir.
- Bir WebSocket bağlantısında yalnızca tek bir defa kullanıcı girişi yapılması yeterlidir.
- WebSocket bağlantısı koparsa, tekrar bağlantı kurulduğunda yeniden kullanıcı girişi paketi gönderilmelidir.
- Bir defa kullanıcı girişi yapıldıktan sonra, tekrar kullanıcı girişi yapılması desteklenmez. Ancak daha önceden misafir giriş bildirimi yapıldıysa, aynı bağlantı üzerinden kullanıcı girişi yapılabilir.
-

Model Formatı

Modeller, Sunucu ile istemci arasındaki gönderilen ve alınan tüm veri tipleridir. Gelen ve gönderilen verinin hangi modele ait olduğu modelin tipi ile belirlenir. Model tipleri sayısal (integer) değerlerdir. Her model objesinde bir tip mutlaka olmalıdır.

Model formatı, type safe/unsafe tüm programlama dillerinde kolay kullanım sağlaması amacıyla aşağıdaki şekilde tanımlanmıştır.

```
[ model_type , model_object ]
```

Yukarıdaki formata göre model tipi sayısal bir değer olmalıdır. Model objesi dizi olmayan herhangi bir nesne olabilir. Model objesindeki tek koşul “type” adında bir sayısal değişkeni olmalı ve bu değişkenin değeri model tipine eşit olmalıdır.

Javascript’te bu modeli oluşturmak için aşağıdaki şekilde bir kod kullanılabilir.

```
var model = [  
    101,  
    {  
        type: 101,  
        username: 'test',  
        token: '123'  
    }  
];
```

C# gibi tip güvenliği olan programlama dillerinde bu model object array olarak okunduktan sonra array’in ilk elemanını integer’a cast ederek ikinci elemanın (modelin kendisinin) tipini bu integer değerden okuyabilirsiniz. Bu değere göre array’in ikinci elemanı olan modeli, olması gereken tipe deserialize edebilirsiniz.

Yukarıdaki örnek Javascript kodunda gösterilen model, JSON formatında serialize edilerek WebSocket üzerinden string data olarak gönderilir.

Bu format, hem istemciden sunucuya, hem de sunucudan istemciye veri gönderilirken aynı şekilde kullanılır.

Kanallar ve Olaylar

Emir defteri, ticker, son işlemler gibi veriler, bağlantı kurulduktan veya kullanıcı girişi yapıldıktan sonra doğrudan gönderilmez. Her işlem için belirli bir kanal ve olay vardır. Bu kanal ve olaylara abone olunarak bu veriler alınır. Bir kanala bir defa abone olunduktan sonra, abonelikten çıkış mesajı gönderilmediği sürece sunucu bu kanaldaki olayları, istemciye sürekli gönderir.

Sunucuda kanallar genel olarak işlemin gerçekleştiği yerleri (emir defteri, son işlemler gibi) olaylar ise işlemin gerçekleştiği pair değerini (BTCTRY, ETHTRY, XLMUSDT gibi) temsil eder. Bazı özel durumlarda bu değerler değişebilir. Örneğin tüm pairlere ait ticker verisini almak için **ticker** kanalının **all** olayına abone olmak gerekir.

Sunucu 3 kanal üzerinden çalışır:

- 1) **orderbook**: Emir defterindeki işlemleri gönderir. Olay adı pair'in upper-case adıdır. Örn: BTCTRY
- 2) **trade**: Son işlemlere ait verileri gönderir. Veri tipine göre gelen veriler tek tek veya toplu olabilir. Olay adı ise pair'in upper-case adıdır. Örn: BTCTRY
- 3) **ticker**: Güncel ticker verisini gönderir. İki tip olay vardır. **all** olayında, tüm pair'lere ait veriler gelir. Yalnızca belirli bir pair'e ait olayları almak için upper-case pair adı olayına abone olunmalıdır. Örn: BTCTRY

Olaya abone olmak için örnek kod aşağıdaki gibidir:

```
[
  151,
  {
    type:151,
    channel: 'trade',
    event: 'ETHTRY',
    join: true
  }
]
```

Benzer şekilde, abone olunmuş olaydan çıkmak için aşağıdaki gibi bir mesaj gönderilir.

```
[
  151,
  {
    type:151,
    channel: 'ticker',
    event: 'all',
    join: false
  }
]
```

Modeller

Sunucu ile Client arasındaki desteklenen model tipleri aşağıdaki tablolarda verilmiştir.

- Tablodaki Kod, modelin tip değeridir. (Modelin sol üstündeki numara)
- KısaAdı, değişken, sınıf veya enum adı olarak kullanılabilir (bizim bu şekilde kullandığımız) model kodlarıdır.
- Türler (Modelin sağ üstündeki harf kodu) karışıkları ve kullanım şekilleri şöyledir:
 - o **P**: genel paketlerdir. İçinde type barındırır. Alınması veya gönderilmesi için herhangi bir kanala abone olmaya veya üye girişi yapmaya gerek yoktur.
 - o **U**: Yalnızca giriş yapmış kullanıcılar tarafından erişilen genel paketlerdir. Bu paketlerde de kanal ve abonelik bağlantısı yoktur.
 - o **C**: Yalnızca kanallardaki olaylara abone olduğunda alınan paketlerdir. Bu paketlerin modellerinde **type** değişkeninin yanısıra **channel** ve **event** adında string değişkenler de bulunmaktadır.

100	Result	P
Sunucudan istemciye gönderilen bir mesajdır. Sunucuya gönderilen isteklerden dönen modelsiz sonuçlardır. Örneğin UserLogin, Subscription gibi isteklerde Result modeli döner.		
Type	number	Model tipi
id	number	İşlem yapılan modele ait Id değeri. Bazı Result modellerinde kullanılmaktadır. Kullanılmadığı durumlarda değeri 0 olabilir.
Ok	boolean	İşlemin başarılı olma durumu
message	string	işlem mesajı. Genellikle işleme ait kısa kodlar barındırır. Son kullanıcıya gösterilmemelidir. Örneğin subscription result olarak dönen değer: "join trade:BTCTRY" benzeridir.

101	Request	P
Sunucuya gönderilen ekstra isteklerdir. Örneğin, son işlemler listesi tekrar istenildiğinde Request model gönderilir.		
type	number	Model tipi
code	number	İstek kodudur. Paket tipleriyle mümkün olduğunda aynı şekilde tasarlanmıştır. Bu istek kodları doküman sonunda listelenmiştir. Örneğin istemci, son işlemlerin kendisine tekrar gönderilmesini talep etmek için bu code değerine 421 (son işlemlerin model tipi) göndermelidir.

channel	string	İstek belirli bir kanal üzerinden gönderiliyorsa bu değere kanalın adı yazılmalıdır. Kanal dışı isteklerde null gönderilebilir.
event	string	İstek belirli bir olay üzerinden gönderiliyorsa bu değere olayın adı yazılmalıdır. Olay dışı isteklerde null gönderilebilir.

111	UserLogin	P
İstemciden sunucuya gönderilen bir mesajdır. Kullanıcı girişi modelidir.		
type	number	Model tipi
id	number	Kullanıcının id değeridir. Mutlaka gönderilmelidir. Sunucuda Token'dan okunan değerle eşitliği kontrol edilir. Misafir giriş bildirimi yapılmak istenirse bu değere 0 göndermek yeterlidir. (Sistemde misafir giriş bildirimi zorunlu değildir)
username	string	Kullanıcı adı. BtcTurk için e-posta adresidir. Misafir giriş bildirimlerinde "guest" yazılmalıdır.
token	string	Kullanıcının SSO'dan alınmış güncel token değeridir. Misafir giriş bildirimlerinde null gönderilmelidir.

112	UserLoginResult	P
Sunucudan istemciye gönderilen bir mesajdır. Kullanıcı giriş modelinin cevabıdır. Result modeli ile tamamen aynıdır. Yalnızca model tipi 112 şeklindedir.		

113	UserLogout	U
İstemciden sunucuya, cevap olarak da sunucudan istemciye geri gönderilen bir mesajdır. Kullanıcı çıkış modelidir. Bu model alındıktan sonra sunucu, socket bağlantısını sonlandıracaktır.		
type	number	Model tipi
id	number	Kullanıcının Id değeri

151	Subscription	P
İstemciden sunucuya gönderilen bir mesajdır. Bir kanaldaki bir olaya abone olmak için veya bu abonelikten çıkmak için gönderilen modeldir. Abone olunan olaylarla ilgili mesajlar kullanıcıya sürekli gönderilir.		

type	number	Model tipi
channel	string	Kanalın adı. Örneğin; trade, orderbook, ticker vs.
event	string	olayın adı. Olay isimleri genellikle “all” veya upper-case olarak pair adı (Örn: BTCTRY, ETHTRY) şeklinde kullanılır.
join	boolean	true ise abone olunmak isteniyor, false ise abonelikten çıkılıyor demektir.

201	BalanceUpdated	U
Kullanıcının bakiyesi herhangi bir nedenden dolayı (confirmation, para çekme talebi, para yatırma talebi, talep onayları vs) değiştiği zaman, sunucu tarafından gönderilen modeldir. Bu mesaj alındığı zaman kullanıcının bakiyesinin API endpoint’ten güncellenmesi gerekmektedir.		
type	number	Model tipi
userId	number	Kullanıcının Id değeri
payload	string	İşlem açıklaması. Bilgilendirme amaçlıdır.

401	TickerAll	C
Sunucudan istemciye gönderilen bir mesajdır. Tüm Pair’lere ait, tüm ticker verisinin modelidir.		
type	number	Model tipi
items	TickerPair []	TickerPair tipinden bir dizidir. Tek fark olarak her TickerPair modeli içinde type değişkeni bulunmaz. TickerPair modeli aşağıda açıklanmıştır.

402	TickerPair	C
Sunucudan istemciye gönderilen bir mesajdır. Yalnızca belirli bir Pair’e ait tüm ticker verisinin modelidir.		
type	number	Model tipi
PS	string	Pair Symbol. Örneğin; BTCTRY
H	string	High. Son 24 saatteki en yüksek fiyat.
L	string	Low. Son 24 saatteki en düşük fiyat.

LA	string	Last. Son işlemin fiyatı.
O	string	Open. Günün ilk işleminin fiyatı
V	string	Volume. Hacim.
AV	string	Average. Ortalama fiyat.
D	string	Daily. Günlük değişim tutarı.
DS	string	Denominator Symbol. Örneğin BTCTRY için TRY.
NS	string	Numerator Symbol. Örneğin BTCTRY için BTC.
PId	number	Pair Id değeri.
NPr	number	Numerator precision değeri. Örneğin BTC için 8
DPr	number	Denominator precision değeri. Örneğin TRY için 2
Ord	number	Pair'in gösterim sırası.
B	string	Bid. En iyi alış emri fiyatı.
A	string	Ask. En iyi satış emri fiyatı.
DP	string	Daily percent. 24 Saatlik değişim oranıdır.

411	AskBidAll (ŞUAN KULLANILMAMAKTADIR)	C
Sunucudan istemciye gönderilen bir mesajdır. Tüm Pair'lere ait, Ask, Bid, Last verilerini içeren modeldir.		
type	number	Model tipi
items	AskBidPair []	AskBidPair tipinden bir dizidir. Tek fark olarak her AskBidPair modeli içinde type değişkeni bulunmaz. AskBidPair modeli aşağıda açıklanmıştır.

412	AskBidPair (ŞUAN KULLANILMAMAKTADIR)	C
Sunucudan istemciye gönderilen bir mesajdır. Yalnızca belirli bir Pair'e ait Ask, Bid, Last verilerinin içeren modeldir.		
type	number	Model tipi

A	string	Ask. En iyi satış emrinin fiyatı.
B	string	Bid. En iyi alış emrinin fiyatı.
LA	string	Last. Son işlem fiyatı.
PS	string	Pair symbol. Örneğin; BTCTRY

421	TradeList	C
Sunucudan istemciye gönderilen bir mesajdır. Belirtilen Pair'e ait tüm son işlemler listesidir.		
type	number	Model tipi
channel	string	Mesajın kanalı
event	string	Mesajın iletildiği olay adı.
symbol	string	Pair symbol. Örneğin; BTCTRY
items	TradeSingle []	Listedeki bir bir işlem kaydını barındıran dizidir. TradeSingle modeli aşağıda açıklanmıştır. TradeList modelinin içindeki TradeSingle modelleri PS ve type değişkenlerini barındırmaz.

422	TradeSingle	C
Sunucudan istemciye gönderilen bir mesajdır. Belirtilen Pair'e ait, gerçekleşmiş tek bir trade verisidir. Bu veri, işlem gerçekleştiği gibi gönderilir.		
type	number	Model tipi
PS	number	Pair Symbol. Örneğin; BTCTRY.
A	string	Amount. İşlemin miktarı.
D	number	Date. İşlem tarihi. Milisaniye cinsinden UnixTime değeridir.
P	string	Price. İşlemin yapıldığı fiyat.
I	string	İşleme ait benzersiz rakamdır. (Benzersizlik durumu farklı yalnızca kendi pairi içinde geçerlidir)

431	OrderBookFull	C
Sunucudan istemciye gönderilen bir mesajdır. Belirli bir Pair'e ait tüm emir defteri modelidir.		
type	number	Model tipi
CS	number	Changeset. Değişim numarası. Ardışık değerler şeklinde gelir. Şuan için herhangi bir kontrolü yapılmamaktadır.
PS	string	Pair Symbol değeri. Örneğin; BTCTRY
AO	Order []	Satış emirleri listesi. Order nesnesi iki değişken barındırır: A, amount, işlem miktarıdır. P ise price, işlem fiyatıdır.
BO	Order []	Alış emirleri listesi.

432	OrderBookDifference	C
Sunucudan istemciye gönderilen bir mesajdır. Belirli bir Pair'e ait yalnızca değişimlerin olduğu emir defteri verisini gönderir. Kullanıcı bu modeli kullanmak için önceki güncel emir defteri verisine sahip olmalıdır. Bu veri, ilgili olaya abone olunduğunda sunucudan otomatik olarak gönderilir.		
ChangingOrder <ul style="list-style-type: none"> • PS : Emir işlem tipidir. 0: Güncellendi, 1: Yeni eklendi, 3: Silindi • P : Emir price değeridir. • A : Emir miktarıdır. 		
type	number	Model tipi
CS	number	Changeset. Değişim numarası. Ardışık değerler şeklinde gelir. Şuan için herhangi bir kontrolü yapılmamaktadır.
PS	string	Pair Symbol değeri. Örneğin; BTCTRY
AO	ChangingOrder []	Satış emirleri listesi. Order nesnesi iki değişken barındırır: A, amount, işlem miktarıdır. P ise price, işlem fiyatıdır.
BO	ChangingOrder []	Alış emirleri listesi.

441	UserOrderMatch	U
Bağlı kullanıcının bir emri gerçekleştiğinde sunucudan kullanıcıya gönderilen, bu işlemin detaylarını içeren modeldir.		

type	number	Model tipi
id	number	İşlem yapılan modele ait Id değeri.
user	boolean	İşlemi yapılan kullanıcının id değeri.
isBid	string	İşlem alış ise true, satış ise false.
method	number	işlem metodu. 0: limit, 1: market, 2:stop limit
pair	number	işlemin pair id değeri
symbol	string	işlemin pair symbol değeri. Örn; BTCTRY
numTotal	string	Emrin toplam Numerator miktarı
numLeft	string	Emrin, trade gerçekleştikten sonra kalan numerator miktarı.
denomTotal	string	Emrin toplam denominator miktarı
denomLeft	string	Emrin, trade gerçekleştikten sonra kalan denominator miktarı.
amount	string	İşlemin miktarı.
price	string	İşlemin gerçekleştiği fiyat.

451	OrderInsert	U
<p>Sunucudan istemciye gönderilen bir mesajdır. Sisteme kullanıcıya ait bir emir eklendiğinde, kullanıcıya özel gönderilen modeldir. Model ile yeni eklenen emir gönderilecektir. Sunucudan bu model alındığında kullanıcının emirlerinin endpoint'ten güncellenmesi de yapılabilir.</p>		
type	number	Model tipi
pairId	number	Emrin pair id değeri.
symbol	string	Emrin pair symbol değeri.
id	number	Emir Id değeri.
method	number	Emrin tipi (0: limit, 1: market, 2: stop limit)
userId	number	Emri veren kullanıcının Id değeri.
price	string	Emrin fiyatı
amount	string	Emrin miktarı

452	OrderDelete	U
Sunucudan istemciye gönderilen bir mesajdır. Sistemden kullanıcıya ait bir emir silindiğinde, kullanıcıya özel gönderilen modeldir. Model ile silinen emir gönderilecektir. Sunucudan bu model alındığında kullanıcının emirlerinin endpoint'ten güncellenmesi de yapılabilir. Model, OrderInsert modeli ile aynıdır.		

453	OrderUpdate	U
Sunucudan istemciye gönderilen bir mesajdır. Sistemde kullanıcıya ait bir emir güncellendiğinde, kullanıcıya özel gönderilen modeldir. Model ile güncellenen emir gönderilecektir. Sunucudan bu model alındığında kullanıcının emirlerinin endpoint'ten güncellenmesi de yapılabilir. Model, OrderInsert modeli ile aynıdır.		

801	SystemMessage	P
Sistem tarafından, tüm kullanıcılara mesaj gönderildiği zaman iletilen modeldir. Bu model alındığı zaman, uygulamada bir modal/lightbox vs ile kullanıcıya mesaj gösterilmelidir. SystemMessage modeli, ResultModel ile aynı tasarlanmıştır.		
Type	number	Model tipi
id	number	Standart mesaj için sıfırdır. 802 değeri (SitePurge model numarası) gönderildiğinde, kullanıcıya gösterilen modal'da sayfayı yeni butonunun eklenmesi ve tıklanığında SitePurge ile aynı işlemi yapması gerekmektedir.
Ok	boolean	Kullanılmamaktadır. Varsayılan false gönderilir.
message	string	Kullanıcıya gösterilecek mesaj metnidir.

802	SitePurge	P
Bu mesaj alındığında, kullanıcıya herhangi bir mesaj göstermeksizin, doğrudan tüm cache verisinin temizlenmesi ve sayfanın yenilenmesi gerekmektedir.		
Type	number	Model tipi
id	number	Kullanılmamaktadır. Varsayılan 0 gönderilir.
Ok	boolean	Kullanılmamaktadır. Varsayılan false gönderilir.
message	string	Kullanılmamaktadır. Varsayılan null gönderilir.

802	LogoutUser	U
<p>Bu mesaj, oturum açmış kullanıcılara, kullanıcı bazlı gönderilir. Mesaj alındığında kullanıcıya sormaksızın, kullanıcının token ve oturum bilgisi temizlenmelidir. Genellikle şüpheli kullanıcı işlemlerinde kullanıcıyı sistemden çıkarmak için kullanılır. Bu sebeple kullanıcı onayı, zaman aşımı, bekletme gibi algoritmalar kullanılmamalı, işlem en kısa sürede doğrudan gerçekleştirilmelidir.</p>		
Type	number	Model tipi
id	number	Kullanıcının Id değeri.
Ok	boolean	Kullanılmamaktadır. Varsayılan false gönderilir.
message	string	Kullanılmamaktadır. Varsayılan null gönderilir.

Kanal, Olay ve Model Tablosu

WebSocket'in desteklenmediği tarayıcılarda veya platformlarda kullanılmak üzere, WebSocket üzerinden gönderilen verilere belirli endpoint'ler ile de erişmek mümkündür.

Kanal	Olay	Model	Açıklama
ticker	all	401	Belirli periyodlarla (en geç saniyede bir) veri gönderilir. Tüm pair ticker verilerini içerir.
ticker	[PAIR]	402	Belirli periyodlarla (en geç saniyede bir) veri gönderilir.
trade	[PAIR]	421 422	Abone olduğu gibi tek seferlik 421 nolu full trade modeli gönderilir. Daha sonra yalnızca 422 nolu yeni gelen trade modelleri gönderilir.
orderbook	[PAIR]	431	Belirli periyodlarla (en geç saniyede bir) tüm emir defteri verisi gönderilir.
obdiff	[PAIR]	431 432	Abone olduğu gibi tek seferlik 431 nolu full orderbook modeli gönderilir. Daha sonra yalnızca 432 nolu değişim modeli gönderilir.

TradingView

TradingView verisi **tradeview** kanalı üzerinden iletilir. Bu kanalda **PAIR_RESOLUTION** olaylarına abone olunmalıdır.

TradingView datası için desteklenen resolution değerleri 4 saatliğe kadar olanlar dakika cinsinden karşılığı ile **1, 5, 15, 30, 60, 240** şeklindedir. Günlük, haftalık ve aylık olanlar ise **1D, 1W, 1M** şeklindedir.

Örneğin, saatlik BTCTRY trading view datasını takip etmek isteyen bir kullanıcı **tradeview** kanalının **BTCTRY_60** olayına abone olmalıdır.

Başka bir örnekle, günlük ETHTRY trading view datası takip etmek istenildiğinde **tradeview** kanalının **ETHTRY_1D** olayına abone olunmalıdır.

NOT: TradingView için WebSocket üzerinden yalnızca son sütun verisi gönderilir. Örneğin saat 14:40:50 iken dakikalık data 14:40 ile 14:41 arasındaki veriyi, saatlik data 14:00 ile 15:00 arasındaki veriyi temsil eder. Gönderilen data içinde, ilgili verinin hangi zaman dilimine ait olduğunu belirten UNIX timestamp değeri bulunur.

428	TradingView		C
Sunucudan istemciye gönderilen bir mesajdır. Olayda belirtilen pair ve resolution’a ait güncel mum verisini içerir.			
type	number	Model tipi	
D	number	UNIX timestamp, saniye cinsinden.	
P	string	Pair. Örneğin BTCTRY	
R	string	Resolution. Örneğin 1W.	
O	String	Open. Datanın açılış değeri.	
H	String	High. Datanın ulaştığı en yüksek değer.	
L	String	Low. Datanın ulaştığı en düşük değer.	
C	String	Close. Datanın aldığı son değer. (Ticker last)	
V	string	Volume. Belirtilen süre içindeki toplam hacim.	

Doğrudan HTTP EndPoint'leri ile İletişim

WebSocket'in desteklenmediği tarayıcılarda veya platformlarda kullanılmak üzere, WebSocket üzerinden gönderilen verilere belirli endpoint'ler ile de erişmek mümkündür.

EndPoint'lere gönderilen HTTP Request'lerine dönen değerler BTC Trader socket server tarafından beslenen son verilerdir. Örneğin, ticker verisi ortalama saniyede bir güncelleniyor ise, aynı saniye içinde gönderilen tüm istekler, aynı sonucu dönecektir.

Emir Defteri

Emir Defteri için her pair'e ait bir endpoint bulunmaktadır. İsteklerin gönderileceği URL aşağıdaki gibidir:

GET <https://ws-feed-sandbox.btctrader.com/orderbook/{PAIR}>

Pair, parametresinin formatı BTCTRY, ETHTRY, XRPUSDT şeklinde olmalıdır. Örneğin

<https://ws-feed-sandbox.btctrader.com/orderbook/BTCTRY>

Güncel BTC/TRY Emir defteri modelini gönderir.

Dönen sonuç yukarıdaki modellerde belirtilen 431 numaralı OrderBookFull kod isimli modelin JSON formatındadır.

Ticker

Ticker verisi, tüm pair'lere ait ticker verisi döndüren tek bir endpoint'ten oluşur. Aşağıdaki şekilde alınır.

GET <https://ws-feed-sandbox.btctrader.com/ticker/all>

Dönen sonuç yukarıdaki modellerde belirtilen 401 numaralı TickerAll kod isimli modelin Items değişkenidir. (Modelin kendisi değil)

Ticker verisinin user-friendly bir HTML tasarım versiyonu görüntülenmek istenildiğinde aşağıdaki endpoint kullanılabilir.

GET <https://ws-feed-sandbox.btctrader.com/ticker/html>

Yalnızca belirli bir pair'e ait ticker verisini almak için ise aşağıdaki endpoint kullanılabilir. Aşağıdaki link BTCTRY ile örneklenmiştir. Sistemdeki mevcut diğer pair ticker verileri için buradaki parametreyi değiştirebilirsiniz.

GET <https://ws-feed-sandbox.btctrader.com/ticker/pair/BTCTRY>

Son İşlemler

Son işlemler, belirtilen bir pair'e ait işlemleri döndürür. Kullanıcı en son kaç işlemi istediğini parametre ile belirtebilir. Son işlemlerdeki maksimum kayıt sayısı 50 olarak belirlenmiştir. 50'den büyük değer girildiğinde yine son 50 işlem gönderilecektir. Son işlemler için istek aşağıdaki şekilde gönderilmelidir.

GET <https://ws-feed-sandbox.btctrader.com/trade/last/{PAIR}/{COUNT}>

Pair değerine BTCTRY şeklinde pair girilmelidir. Count değerine ise sayısal olarak 1 ila 50 arasında bir değer gönderilmelidir. Count değeri gönderilmezse varsayılan olarak 50 kabul edilir. Örneğin aşağıdaki şekilde istekler geçerlidir.

<https://ws-feed-sandbox.btctrader.com/trade/last/BTCTRY/10>

<https://ws-feed-sandbox.btctrader.com/trade/last/EthTRy/76>

<https://ws-feed-sandbox.btctrader.com/trade/last/xrptry>