Indexer Websocket Documentation

dYdX offers a WebSocket API for streaming v4 updates.

- Forthe deployment by DYDX token holders
- , use wss://indexer.dydx.trade/v4/ws
- ForTestnet
- , use wss://indexer.dydx.trade/v4/ws

Note: Messages on Indexer WebSocket feeds are typically more recent than data fetched via Indexer's REST API, because the latter is backed by read replicas of the databases that feed the former. Ordinarily this difference is minimal (less than a second), but it might become prolonged under load. Please see[Indexer Architecture(opens in a new tab)] for more information.

Overall

Connect

Upon connecting to v4 Websockets you will receive an initial connection message with the following format:

{ "type" : "connected" , "connection_id" : "004a1efa-21bb-4b19-a2e9-a8ffadd6dc53" , "message_id" : 0 }

Maintaining a Connection

Every 30 seconds, the websocket API will send a ping event to the connected client. If a pong event is not received within 10 seconds back, the websocket API will disconnect.

Subscribe

You may subscribe to any channel following the subscribe instructions above. Subscribing to a channel has the following fields:

- type
-
    - Should always be specified tosubscribe
- channel
-
    - Specifies the channel you are subscribing to. The specific string is specified in each channel's documentation
- id
-
    - required for all channels other than market. Specifies the market or subaccount you are subscribing to.

Unsubscribe

Utilize the same message to subscribe but replace the type withunsubscribe . For example:{ "type": "unsubscribe", "channel": "v4_trades", "id": "BTC-USD" }

Example

Use a command-line websocket client such as[interactive-websocket-cli(opens in a new tab)] to connect and subscribe to channels.

Example (withinteractive-websocket-cli)

For the deployment by

DYDX token holders , use

wscli connect wss

: //indexer.dydx.trade/v4/ws wscli connect wss : //indexer.v4testnet.dydx.exchange/v4/ws < output

from

ws-cli

< type

's'

to

send

{ "type" : "subscribe" ,

"channel" : "v4_trades" ,

"id" : "BTC-USD" }

Subaccounts

This channel provides realtime information about orders, fills, transfers, perpetual positions, and perpetual assets for a subaccount.

Subscribe

Field Type Description type string Set to subscribe channel string Set to v4_subaccounts id string Set to the address and subaccount number in the format {address}/{subaccount_number}

Initial Response

Returns everything from the/v4/addresses/:address/subaccountNumber/:subaccountNumber , and/v4/orders?addresses={address}&subaccountNumber={subaccountNumber}&status=OPEN .

Example

{ "type" : "subscribed" , "connection_id" : "c5a28fa5-c257-4fb5-b68e-fe084c2768e5" , "message_id" : 1 , "channel" : "v4_subaccounts" , "id" : "dydx199tqg4wdlnu4qjlxchpd7seg454937hjrknju4/0" , "contents" : { "subaccount" : { "address" :

"dydx199tqg4wdlnu4qjlxchpd7seg454937hjrknju4" , "subaccountNumber" :

0 , "equity" :

"100000000000.000000" , "freeCollateral" :

"100000000000.000000" , "openPerpetualPositions" : {} , "assetPositions" : { "USDC" : { "symbol" :

"USDC" , "side" :

"LONG" , "size" :

"100000000000" , "assetId" :

"0" } } , "marginEnabled" :

true } , "orders" : [] } }

Channel Data

Subsequent responses will contain any update to open orders, changes in account, changes in open positions, and/or transfers in a single message.

export

interface

SubaccountsChannelData { channel :

'v4_trades' , id :

string , contents :

SubaccountMessageContents , blockHeight :

string , transactionIndex :

number , eventIndex :

number , clobPairId :

string , version :

string , }

export

interface

SubaccountMessageContents { // Perpetual position updates on the subaccount perpetualPositions ?:

PerpetualPositionSubaccountMessageContents [] , // Asset position updates on the subaccount assetPositions ?:

AssetPositionSubaccountMessageContents [] , // Order updates on the subaccount orders ?:

OrderSubaccountMessageContents [] , // Fills that occur on the subaccount fills ?:

FillSubaccountMessageContents [] , // Transfers that occur on the subaccount transfers ?:

TransferSubaccountMessageContents , }

export

interface

PerpetualPositionSubaccountMessageContents { address :

string , subaccountNumber :

number , positionId :

string , market :

string , side :

PositionSide , status :

PerpetualPositionStatus , size :

string , maxSize :

string , netFunding :

string , entryPrice :

string , exitPrice ?:

string , sumOpen :

string , sumClose :

string , realizedPnl ?:

string , unrealizedPnl ?:

string , }

export

enum

PositionSide { LONG =

'LONG' , SHORT =

'SHORT' , }

export

enum

PerpetualPositionStatus { OPEN =

'OPEN' , CLOSED =

'CLOSED' , LIQUIDATED =

'LIQUIDATED' , }

export

interface

AssetPositionSubaccountMessageContents { address :

string , subaccountNumber :

number , positionId :

string , assetId :

string , symbol :

string , side :

PositionSide , size :

string , }

export

interface

OrderSubaccountMessageContents { id :

string ; subaccountId :

string ; clientId :

string ; clobPairId :

string ; side :

OrderSide ; size :

string ; ticker :

string , price :

string ; type :

OrderType ; timeInForce :

APITimeInForce ; postOnly :

boolean ; reduceOnly :

boolean ; status :

APIOrderStatus ; orderFlags :

string ; totalFilled ?:

string ; totalOptimisticFilled ?:

string ; goodTilBlock ?:

string ; goodTilBlockTime ?:

string ; removalReason ?:

string ; createdAtHeight ?:

string ; clientMetadata :

string ; triggerPrice ?:

string ; updatedAt ?:

IsoString ; updatedAtHeight ?:

string ; }

export

enum

OrderSide { BUY =

'BUY' , SELL =

'SELL' , }

export

enum

OrderType { LIMIT =

'LIMIT' , MARKET =

'MARKET' , STOP_LIMIT =

'STOP_LIMIT' , STOP_MARKET =

'STOP_MARKET' , TRAILING_STOP =

'TRAILING_STOP' , TAKE_PROFIT =

'TAKE_PROFIT' , TAKE_PROFIT_MARKET =

'TAKE_PROFIT_MARKET' , HARD_TRADE =

'HARD_TRADE' , FAILED_HARD_TRADE =

'FAILED_HARD_TRADE' , TRANSFER_PLACEHOLDER =

'TRANSFER_PLACEHOLDER' , }

export

enum

APITimeInForce { // GTT represents Good-Til-Time, where an order will first match with existing orders on the book // and any remaining size will be added to the book as a maker order, which will expire at a // given expiry time. GTT =

'GTT' , // FOK represents Fill-Or-KIll where it's enforced that an order will either be filled // completely and immediately by maker orders on the book or canceled if the entire amount can't // be filled. FOK =

'FOK' , // IOC represents Immediate-Or-Cancel, where it's enforced that an order only be matched with // maker orders on the book. If the order has remaining size after matching with existing orders // on the book, the remaining size is not placed on the book. IOC =

'IOC' , }

export

enum

APIOrderStatus { OPEN =

'OPEN' , FILLED =

'FILLED' , CANCELED =

'CANCELED' , BEST_EFFORT_CANCELED =

'BEST_EFFORT_CANCELED' , BEST_EFFORT_OPENED =

'BEST_EFFORT_OPENED' , UNTRIGGERED =

"UNTRIGGERED" }

export

interface

FillSubaccountMessageContents { id :

string ; subaccountId :

string ; side :

OrderSide ; liquidity :

Liquidity ; type :

FillType ; clobPairId :

string ; size :

string ; price :

string ; quoteAmount :

string ; eventId :

string , transactionHash :

string ; createdAt :

IsoString ; createdAtHeight :

string ; orderId ?:

string ; ticker :

string ; }

export

enum

Liquidity { TAKER =

'TAKER' , MAKER =

'MAKER' , }

export

enum

FillType { // LIMIT is the fill type for a fill with a limit taker order. LIMIT =

'LIMIT' , // LIQUIDATED is for the taker side of the fill where the subaccount was liquidated. // The subaccountId associated with this fill is the liquidated subaccount. LIQUIDATED =

'LIQUIDATED' , // LIQUIDATION is for the maker side of the fill, never used for orders LIQUIDATION =

'LIQUIDATION' , // DELEVERAGED is for the subaccount that was deleveraged in a deleveraging event. // The fill type will be set to taker. DELEVERAGED =

'DELEVERAGED' , // OFFSETTING is for the offsetting subaccount in a deleveraging event. // The fill type will be set to maker. OFFSETTING =

'OFFSETTING' , }

export

enum

TradeType { // LIMIT is the trade type for a fill with a limit taker order. LIMIT =

'LIMIT' , // LIQUIDATED is the trade type for a fill with a liquidated taker order. LIQUIDATED =

'LIQUIDATED' , // DELEVERAGED is the trade type for a fill with a deleveraged taker order. DELEVERAGED =

'DELEVERAGED' , }

export

interface

TransferSubaccountMessageContents { sender : { address :

string , subaccountNumber ?:

number , } , recipient : { address :

string , subaccountNumber ?:

number , } , symbol :

string , size :

string , type :

TransferType , createdAt :

IsoString , createdAtHeight :

string , transactionHash :

string , }

export

enum

TransferType { TRANSFER_IN =

'TRANSFER_IN' , TRANSFER_OUT =

'TRANSFER_OUT' , DEPOSIT =

'DEPOSIT' , WITHDRAWAL =

'WITHDRAWAL' , }

Example

{ "type" : "channel_data" , "connection_id" : "a00edbe8-095a-4da1-8a9d-ff1f91467258" , "message_id" : 4 , "id" : "dydx1zsw8fczav25uvc8rg3zcv6zy9j7yhnktpq374m/0" , "channel" : "v4_subaccounts" , "version" : "2.1.0" , "contents" : { "orders" : [{ "id" :

"64fe30a2-006d-5108-a156-cb0c8443546c" , "side" :

"BUY" , "size" :

"1" , "totalFilled" :

"1" , "price" :

"1948.65" , "type" :

"LIMIT" , "status" :

"FILLED" , "timeInForce" :

"IOC" , "reduceOnly" :

false , "orderFlags" :

"0" , "goodTilBlock" :

"61186" , "goodTilBlockTime" :

null , "postOnly" :

false , "ticker" :

"ETH-USD" }] , "fills" : [{ "id" :

"c5030bd3-cd85-5046-8f2a-518bbba6ec45" , "subaccountId" :

"db535c19-b298-5ee8-bb59-e96c659a8bd4" , "side" :

"BUY" , "liquidity" :

"TAKER" , "type" :

"LIMIT" , "clobPairId" :

"1" , "orderId" :

"64fe30a2-006d-5108-a156-cb0c8443546c" , "size" :

"1" , "price" :

"1854.25" , "quoteAmount" :

"1854.25" , "eventId" : { "type" :

"Buffer" , "data" : [0 , 0 , 238 , 241 , 0 , 0 , 0 , 2 , 0 , 0 , 0 , 77] } , "transactionHash" :

"C84B0BBCA8E713A2D46EFBA07F2D0A32C1F6E2440794A366B888503935E0EF40" , "createdAt" :

"2023-04-04T19:09:24.869Z" , "createdAtHeight" :

"61169" , "ticker" :

"ETH-USD" }] } }

Orderbooks

Subscribe

Field Type Description type string Set to subscribe channel string Set to v4_orderbook id string Set to the ticker of the market you would like to subscribe to. For example, BTC-USD

Initial Response

Returns everything fromv4/orderbooks/perpetualMarkets{id} endpoint.

- Example
- {
- "type"
- :
- "subscribed"
- ,
- "connection_id"
- :
- "ee5a4696-dce8-44ef-8d68-f0e0d0b06160"
- ,
- "message_id"
- :
- 2
- ,
- "channel"
- :
- "v4_orderbook"
- ,
- "id"
- :
- "BTC-USD"
- ,
- "contents"

- : {
- "bids"
- :
- [
- {
- "price"
- :
- "28194"
- ,
- "size"
- :
- "4.764826096"
- }
- ,
- {
- "price"
- :
- "28193"
- ,
- "size"
- :
- "3.115323739"
- }
- ,
- {
- "price"
- :
- "28192"
- ,
- "size"
- :
- "3.400340775"
- }
- ,
- {
- "price"
- :
- "28191"
- ,
- "size"
- :
- "3.177700682"
- }
- ,
- {
- "price"
- :
- "28190"
- ,
- "size"
- :
- "3.055502176"
- }
- ,
- {
- "price"
- :
- "28189"
- ,
- "size"
- :
- "3.672892171"
- }
- ,
- {
- "price"
- :
- "28188"

- ,
- "size"
- :
- "3.597672948"
- }
- ,
- {
- "price"
- :
- "28187"
- ,
- "size"
- :
- "2.561597964"
- }
- ,
- {
- "price"
- :
- "28186"
- ,
- "size"
- :
- "3.070490554"
- }
- ,
- {
- "price"
- :
- "28185"
- ,
- "size"
- :
- "3.550128411"
- }
- ,
- {
- "price"
- :
- "28184"
- ,
- "size"
- :
- "4.213369101"
- }
- ,
- {
- "price"
- :
- "28183"
- ,
- "size"
- :
- "3.608880877"
- }
- ,
-]
- ,
- "asks"
- :
- [
- {
- "price"
- :
- "28195"
- ,
- "size"
- :

- "3.219612343"
- }
- ,
- {
- "price"
- :
- "28196"
- ,
- "size"
- :
- "2.387087565"
- }
- ,
- {
- "price"
- :
- "28197"
- ,
- "size"
- :
- "2.698530469"
- }
- ,
- {
- "price"
- :
- "28198"
- ,
- "size"
- :
- "2.590884421"
- }
- ,
- {
- "price"
- :
- "28199"
- ,
- "size"
- :
- "3.192796678"
- }
- ,
-]
- ,
- }
- ,
- }

Channel Data

interface

OrderbookChannelData { channel :

'v4_orderbook' , id :

string , contents :

OrderbookMessageContents , clobPairId :

string , version :

string , }

interface

OrderbookMessageContents { bids ?:

PriceLevel [] , asks ?:

PriceLevel [] , }

// The first string indicates the price, the second string indicates the size type

PriceLevel

= [string ,

string]; * Example * { * "type" * : * "channel_data" * , * "connection_id" * : * "ee5a4696-dce8-44ef-8d68-f0e0d0b06160" * , * "message_id" * : * 484 * , * "id" * : * "BTC-USD" * , * "channel" * : * "v4_orderbook" * , * "version" * : * "0.0.1" * , * "contents" * : { * "bids" * : * [* [* "27773" * , * "1.986168516" *] *] * } * }

Trades

Subscribe

Field Type Description type string Set to subscribe channel string Set to v4_trades id string Set to the ticker of the market you would like to subscribe to. For example, BTC-USD

Initial Response

Returns everything fromv4/trades/perpetualMarkets{id} endpoint.

- Example
- {
- "type"
- :
- "subscribed"
- ,
- "connection_id"
- :
- "57844645-0b1d-4c3f-ad71-1c6154154a13"
- ,
- "message_id"
- :
- 1
- ,
- "channel"
- :
- "v4_trades"
- ,
- "id"
- :
- "BTC-USD"
- ,
- "contents"
- : {
- "trades"
- :
- [
- {
- "side"
- :
- "BUY"
- ,
- "size"
- :
- "0.00396135"
- ,
- "price"
- :
- "27848"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:56.226Z"
- ,
- "createdAtHeight"

- :
- "49592"
- }
- ,
- {
- "side"
- :
- "SELL"
- ,
- "size"
- :
- "0.000019216"
- ,
- "price"
- :
- "27841"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:56.226Z"
- ,
- "createdAtHeight"
- :
- "49592"
- }
- ,
- {
- "side"
- :
- "SELL"
- ,
- "size"
- :
- "0.001682908"
- ,
- "price"
- :
- "27840"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:56.226Z"
- ,
- "createdAtHeight"
- :
- "49592"
- }
- ,
- {
- "side"
- :
- "SELL"
- ,
- "size"
- :
- "0.000311013"
- ,
- "price"
- :
- "27840"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:56.226Z"
- ,
- "createdAtHeight"
- :
- "49592"

- }
- ,
- {
- "side"
- :
- "SELL"
- ,
- "size"
- :
- "0.000000011"
- ,
- "price"
- :
- "27842"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:56.226Z"
- ,
- "createdAtHeight"
- :
- "49592"
- }
- ,
- {
- "side"
- :
- "SELL"
- ,
- "size"
- :
- "0.000000017"
- ,
- "price"
- :
- "27842"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:56.226Z"
- ,
- "createdAtHeight"
- :
- "49592"
- }
- ,
- {
- "side"
- :
- "SELL"
- ,
- "size"
- :
- "0.000226026"
- ,
- "price"
- :
- "27842"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:56.226Z"
- ,
- "createdAtHeight"
- :
- "49592"
- }
- ,

- {
- "side"
- :
- "SELL"
- ,
- "size"
- :
- "0.000000004"
- ,
- "price"
- :
- "27842"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:56.226Z"
- ,
- "createdAtHeight"
- :
- "49592"
- }
- ,
- {
- "side"
- :
- "SELL"
- ,
- "size"
- :
- "0.000000006"
- ,
- "price"
- :
- "27842"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:56.226Z"
- ,
- "createdAtHeight"
- :
- "49592"
- }
- ,
- {
- "side"
- :
- "SELL"
- ,
- "size"
- :
- "0.000226015"
- ,
- "price"
- :
- "27842"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:56.226Z"
- ,
- "createdAtHeight"
- :
- "49592"
- }
- ,
- {
- "side"

- :
- "SELL"
- ,
- "size"
- :
- "0.000003739"
- ,
- "price"
- :
- "27842"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:56.226Z"
- ,
- "createdAtHeight"
- :
- "49592"
- }
- ,
- {
- "side"
- :
- "SELL"
- ,
- "size"
- :
- "0.000164144"
- ,
- "price"
- :
- "27842"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:56.226Z"
- ,
- "createdAtHeight"
- :
- "49592"
- }
- ,
- {
- "side"
- :
- "BUY"
- ,
- "size"
- :
- "0.037703477"
- ,
- "price"
- :
- "27848"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:50.516Z"
- ,
- "createdAtHeight"
- :
- "49591"
- }
- ,
- {
- "side"
- :
- "SELL"

- ,
- "size"
- :
- "0.000000321"
- ,
- "price"
- :
- "27842"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:50.516Z"
- ,
- "createdAtHeight"
- :
- "49591"
- }
- ,
- {
- "side"
- :
- "SELL"
- ,
- "size"
- :
- "0.06706869"
- ,
- "price"
- :
- "27842"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:50.516Z"
- ,
- "createdAtHeight"
- :
- "49591"
- }
- ,
- {
- "side"
- :
- "SELL"
- ,
- "size"
- :
- "0.002573305"
- ,
- "price"
- :
- "27842"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:50.516Z"
- ,
- "createdAtHeight"
- :
- "49591"
- }
- ,
- {
- "side"
- :
- "SELL"
- ,
- "size"

- :
- "0.001525924"
- ,
- "price"
- :
- "27842"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:50.516Z"
- ,
- "createdAtHeight"
- :
- "49591"
- }
- ,
- {
- "side"
- :
- "SELL"
- ,
- "size"
- :
- "0.00387205"
- ,
- "price"
- :
- "27842"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:50.516Z"
- ,
- "createdAtHeight"
- :
- "49591"
- }
- ,
- {
- "side"
- :
- "SELL"
- ,
- "size"
- :
- "0.000094697"
- ,
- "price"
- :
- "27845"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:50.516Z"
- ,
- "createdAtHeight"
- :
- "49591"
- }
- ,
- {
- "side"
- :
- "SELL"
- ,
- "size"
- :
- "0.002828331"

- ,
- "price"
- :
- "27842"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:50.516Z"
- ,
- "createdAtHeight"
- :
- "49591"
- }
- ,
- {
- "side"
- :
- "SELL"
- ,
- "size"
- :
- "0.000100428"
- ,
- "price"
- :
- "27845"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:50.516Z"
- ,
- "createdAtHeight"
- :
- "49591"
- }
- ,
- {
- "side"
- :
- "BUY"
- ,
- "size"
- :
- "0.000098184"
- ,
- "price"
- :
- "27848"
- ,
- "createdAt"
- :
- "2023-04-04T00:28:50.516Z"
- ,
- "createdAtHeight"
- :
- "49591"
- }
- ,
-]
- ,
- }
- ,
- }

Channel Data

interface

TradeChannelData { channel :

'v4_trades' , id :

string , contents :

TradeMessageContents , blockHeight :

string , clobPairId :

string , version :

string , }

interface

TradeMessageContents { trades :

TradeContent [] , }

interface

TradeContent { // Unique id of the trade, which is the taker fill id. id :

string , size :

string , price :

string , side :

string , createdAt :

IsoString , type :

TradeType , }

Example

{ "type" : "channel_data" , "connection_id" : "57844645-0b1d-4c3f-ad71-1c6154154a13" , "message_id" : 4 , "id" : "BTC-USD" , "channel" : "v4_trades" , "version" : "1.1.0" , "contents" : { "trades" : [{ "id" :

"8ee6d90d-272d-5edd-bf0f-2e4d6ae3d3b7" , "size" :

"0.000100431" , "price" :

"27839" , "side" :

"BUY" , "createdAt" :

"2023-04-04T00:29:19.353Z" , "type" :

"LIQUIDATED" } , { "id" :

"38e64479-af09-5417-a795-195f83879156" , "size" :

"0.000000004" , "price" :

"27839" , "side" :

"BUY" , "createdAt" :

"2023-04-04T00:29:19.353Z" , "type" :

"LIQUIDATED" } , { "id" :

"d310c32c-f066-5ba8-a97d-10a29d9a6c84" , "size" :

"0.000000005" , "price" :

"27837" , "side" :

"SELL" , "createdAt" :

"2023-04-04T00:29:19.353Z" , "type" :

"LIMIT" } , { "id" :

"dd1088b5-5cab-518f-a59c-4d5f735ab861" , "size" :

"0.000118502" , "price" :

"27837" , "side" :

"SELL" , "createdAt" :

"2023-04-04T00:29:19.353Z" , "type" :

"LIMIT" } ,] , } , }

Markets

Subscribe

Field Type Description type string Set to subscribe channel string Set to v4_markets

Initial Response

Returns everything fromv4/perpetualMarkets endpoint.

Example

{ "type" : "subscribed" , "connection_id" : "6e0af39b-5937-459a-b7ac-cc8abe1049db" , "message_id" : 1 , "channel" : "v4_markets" , "contents" : { "markets" : { "BTC-USD" : { "clobPairId" :

"0" , "ticker" :

"BTC-USD" , "status" :

"ACTIVE" , "baseAsset" :

"" , "quoteAsset" :

"" , "oraclePrice" :

"27752.92" , "priceChange24H" :

"0" , "volume24H" :

"63894023.044245577" , "trades24H" :

143820 , "nextFundingRate" :

"0" , "initialMarginFraction" :

"0.050000" , "maintenanceMarginFraction" :

"0.030000" , "basePositionSize" :

"0" , "incrementalPositionSize" :

"0" , "maxPositionSize" :

"0" , "openInterest" :

"1891.473716288" , "atomicResolution" :

- 10 , "quantumConversionExponent" :

- 8 , "tickSize" :

"1" , "stepSize" :

"0.000000001" , "stepBaseQuantums" :

10 , "subticksPerTick" :

10000 } , "ETH-USD" : { "clobPairId" :

"1" , "ticker" :

"ETH-USD" , "status" :

"ACTIVE" , "baseAsset" :

"" , "quoteAsset" :

"" , "oraclePrice" :

"1808.2" , "priceChange24H" :

"0" , "volume24H" :

"67487133.70842158" , "trades24H" :

137552 , "nextFundingRate" :

"0" , "initialMarginFraction" :

"0.050000" , "maintenanceMarginFraction" :

"0.030000" , "basePositionSize" :

"0" , "incrementalPositionSize" :

"0" , "maxPositionSize" :

"0" , "openInterest" :

"44027.853711" , "atomicResolution" :

- 9 , "quantumConversionExponent" :

- 9 , "tickSize" :

"0.01" , "stepSize" :

"0.000001" , "stepBaseQuantums" :

1000 , "subticksPerTick" :

10000 } } } }

Channel Data

interface

MarketChannelData { channel :

'v4_markets' , id :

'v4_markets' , contents :

MarketMessageContents , version :

string , }

interface

MarketMessageContents { trading ?:

TradingMarketMessageContents , oraclePrices ?:

OraclePriceMarketMessageContentsMapping , }

type

TradingMarketMessageContents

= { [ticker :

string] :

TradingPerpetualMarketMessage };

interface

TradingPerpetualMarketMessage { id ?:

string ; clobPairId ?:

string ; ticker ?:

string ; marketId ?:

number ; status ?:

PerpetualMarketStatus ; // 'ACTIVE', 'PAUSED', 'CANCEL_ONLY', 'POST_ONLY', or 'INITIALIZING' baseAsset ?:

string ; quoteAsset ?:

string ; initialMarginFraction ?:

string ; maintenanceMarginFraction ?:

string ; basePositionSize ?:

string ; incrementalPositionSize ?:

string ; maxPositionSize ?:

string ; openInterest ?:

string ; quantumConversionExponent ?:

number ; atomicResolution ?:

number ; subticksPerTick ?:

number ; stepBaseQuantums ?:

number ; priceChange24H ?:

string ; volume24H ?:

string ; trades24H ?:

number ; nextFundingRate ?:

string ; }

type

OraclePriceMarketMessageContentsMapping

= { [ticker :

string] :

OraclePriceMarket , };

interface

OraclePriceMarket { oraclePrice :

string , effectiveAt :

IsoString , effectiveAtHeight :

string , marketId :

number , }

Example

{ "type" : "channel_data" , "connection_id" : "1f4ec0e3-ff95-48cc-94f1-7118a19412ff" , "message_id" : 2 , "channel" : "v4_markets" , "version" : "0.0.1" , "contents" : { "trading" : { "BTC-USD" : { "nextFundingRate" :

"0.00006821875" } , "ETH-USD" : { "volume24H" :

"1462890959.6541" , "nextFundingRate" :

"0.00007478125" } } } } Last updated onFebruary 29, 2024[Indexer API](#) [Indexer Client](#)