

# Backpack Exchange API (1.0)

Download OpenAPI specification: [Download](#)

## Introduction

Welcome to the Backpack Exchange API. This API is for programmatic trade execution. All of the endpoints require requests to be signed with an ED25519 keypair for authentication.

The API is hosted at <https://api.backpack.exchange/> and the WS API is hosted at <wss://ws.backpack.exchange/>.

## Authentication

### Signing requests

Signed requests are required for any API calls that mutate state. Additionally, some read only requests can be performed by signing or via session authentication.

Signed requests require the following additional headers:

- `X-Timestamp` - Unix time in milliseconds that the request was sent.
- `X-Window` - Time window in milliseconds that the request is valid for, default is `5000` and maximum is `60000`.
- `X-API-Key` - Base64 encoded verifying key of the ED25519 keypair.
- `X-Signature` - Base64 encoded signature generated according to the instructions below.

To generate a signature perform the following:

1. The key/values of the request body or query parameters should be ordered alphabetically and then turned into query string format.

2. Append the header values for the timestamp and receive window to the above generated string in the format `&timestamp=<timestamp>&window=<window>`. If no `X-Window` header is passed the default value of `5000` still needs to be added to the signing string.

Each request also has an instruction type, valid instructions are:

```
accountQuery
balanceQuery
borrowLendExecute
borrowHistoryQueryAll
collateralQuery
depositAddressQuery
depositQueryAll
fillHistoryQueryAll
fundingHistoryQueryAll
interestHistoryQueryAll
orderCancel
orderCancelAll
orderExecute
orderHistoryQueryAll
orderQuery
orderQueryAll
pnlHistoryQueryAll
positionQuery
quoteSubmit
strategyCancel
strategyCancelAll
strategyCreate
strategyHistoryQueryAll
strategyQuery
strategyQueryAll
withdraw
withdrawalQueryAll
```

The correct instruction type should be prefixed to the signing string. The instruction types for each request are documented alongside the request.

For example, an API request to cancel an order with the following body:

```
{
  "orderId": 28
  "symbol": "BTC_USDT",
}
```

Would require the following to be signed:

```
instruction=orderCancel&orderId=28&symbol=BTC_USDT&timestamp=1614550000000&wind
```

Regarding batch order execution (`POST /orders`), for each order in the batch, the order parameters should be ordered alphabetically and then turned into query string format. The `orderExecute` instruction

should then be prefixed to that string. The query strings for the orders should be concatenated with `&` and the timestamp and window appended at the end.

For example, an API request for an order execution batch with the following body:

```
[  
  {  
    "symbol": "SOL_USDC_PERP",  
    "side": "Bid",  
    "orderType": "Limit",  
    "price": "141",  
    "quantity": "12"  
  },  
  {  
    "symbol": "SOL_USDC_PERP",  
    "side": "Bid",  
    "orderType": "Limit",  
    "price": "140",  
    "quantity": "11"  
  }  
]
```

Would require the following to be signed:

```
instruction=orderExecute&orderType=Limit&price=141&quantity=12&side=Bid&symbol=SOL_USDC_PERP
```

If the API endpoint requires query parameters instead of a request body, the same procedure should be used on the query parameters. If the API endpoint does not have a request body or query parameters, only the timestamp and receive window need to be signed.

This message should be signed using the private key of the ED25519 keypair that corresponds to the public key in the `X-API-Key` header. The signature should then be base64 encoded and submitted in the `X-Signature` header.

## Changelog

2025-10-23

- Add `j` and `k` fields to the order update stream (take profit limit price and stop loss limit price).

2025-09-02

- The `/depth` endpoint now returns a limit of 5,000 price levels on each side of the book.

2025-09-01

- The `cumulativeInterest` response field is being removed from the `/position` endpoint.
- Estimated liquidation price or `1` is being removed from the position update stream. It will remain as a placeholder and be set to 0. It will be removed in the future, so client's should not rely on its presence.
- Liquidation price can be queried for a single position using the Positions API `/position` for example `/position?symbol=BTC_USDC_PERP`.

2025-08-07

- `/history/pnl` has been removed.

2025-06-08

- The order id format is changing, it is no longer a byte shifted timestamp. It is no longer possible to derive the order timestamp from the order id. This change will take place at Monday June 9th, 01:00 UTC.

2025-04-22

- The `/fills` endpoint now returns all fills for the account, including fills from system orders as well as client orders. System orders include liquidations, ADLs and collateral conversions. Previously, by default, it only returned fills from client orders. This behavior can be achieved by setting the `fillType` parameter to `User`.

2025-04-08

- Added funding rate lower and upper bounds to `/markets` and `/market` endpoints.

2025-03-26

- Add open interest stream `openInterest.<symbol>`.
- Added the option to query `/history/borrowLend/positions` with a signed request using the instruction `borrowPositionHistoryQueryAll`.

2025-03-19

- The leverage filter has been removed from `/markets` and `/market` endpoints.
- Added `/openInterest` now takes `symbol` as an optional parameter. When not set, all markets are returned.
- `/openInterests` has been deprecated.
- Add stop loss and take profit fields to `/orders/execute`.
- Add `I` field to the order update stream (related order id).
- Add `a` and `b` fields to the order update stream (take profit trigger price and stop loss trigger price).

2025-02-28

- Added `clientId` to fill history.

2025-02-11

- An `o` field has been added to the order update stream. It denotes the origin of the update. The possible values are:
  - `USER` : The origin of the update was due to order entry by the user.
  - `LIQUIDATION_AUTOCLOSE` : The origin of the update was due to a liquidation by the liquidation engine.
  - `ADL_AUTOCLOSE` : The origin of the update was due to an ADL (auto-deleveraging) event.
  - `COLLATERAL_CONVERSION` : The origin of the update was due to a collateral conversion to settle debt on the account.
  - `SETTLEMENT_AUTOCLOSE` : The origin of the update was due to the settlement of a position on a dated market.
  - `BACKSTOP_LIQUIDITY_PROVIDER` : The origin of the update was due to a backstop liquidity provider facilitating a liquidation.

2025-02-07

- Added `r` to denote a reduce only order on the order updates stream.
- Added `reduceOnly` to the get orders endpoint.

2025-02-03

- Added `openInterestLimit` to the markets endpoint. Applicable to futures markets only.
- Added `orderModified` event to the order update stream. A resting reduce only order's quantity can be decreased in order to prevent position side reversal.

2025-01-09

- Added `marketType` to the markets endpoint.
- Added an optional `marketType` filter to the fills and the orders endpoints.

2024-12-03

- Add order expiry reason to order update stream.
- Add `cumulativeInterest` to borrow lend position.

2024-12-02

- Add borrow lend history per position endpoint.

2024-11-10

- Add `timestamp` field denoting the system time in unix-epoch microseconds to the depth endpoint.

2024-10-15

- Convert all error responses to JSON and add a error code.

2024-05-14

- Add `executedQuantity` and `executedQuoteQuantity` to order history endpoint.

2024-05-03

- Add single market order update stream `account.orderUpdate.<symbol>`.

2024-05-02

- Add optional `from` and `to` timestamp to get withdrawals endpoint.

2024-05-01

- Add optional `from` and `to` timestamp to get deposits endpoint.

2024-03-14

- Add optional `orderId` filter to order history endpoint.
- Add optional `from` and `to` timestamp to order fills endpoint.

2024-02-28

- Return the withdrawal in request withdrawal response.

2024-02-24

- An additional field `t` was added to the private order update stream. It is the `trade_id` of the fill that generated the order update.
- Added a maximum value for the `X-Window` header of `60000`.

2024-01-16

## Breaking

- A new websocket API is available at `wss://ws.backpack.exchange`. Please see the documentation. The previous API remains on the same endpoint and will be deprecated after a migration period. The new API changes the following:
  - Subscription endpoint is now `wss://ws.backpack.exchange` instead of `wss://ws.backpack.exchange/stream`.
  - Can subscribe and unsubscribe to/from multiple streams by passing more than one in the `params` field.
  - Signature should now be sent in a separate `signature` field.
  - Signature instruction changed from `accountQuery` to `subscribe`.
  - Event and engine timestamps are now in `microseconds` instead of `milliseconds`.
  - Add engine timestamp to `bookTicker`, `depth`, and `order` streams.
  - Add quote asset volume to ticker stream.
  - Add sequential trade id to trade stream.
  - Rename the event type in the depth stream from `depthEvent` to `depth`.
  - Change the format of streams from `<symbol>@<type>` to `<type>.<symbol>` or `kline.<interval>.<symbol>` for K-lines.
  - Flatten the K-Line in the K-line stream so its not nested.

2024-01-11

## Breaking

- Replaced `identifier` field on deposits with `transaction_hash` and `provider_id`. This aims to provide clearer representation of the field, particularly for fiat deposits.

- Removed duplicate `pending` values from the `WithdrawalStatus` and `DepositStatus` spec enum.
- 

## Assets

Assets and collateral data.

### Get assets.

Get all supported assets.

## Responses

› **200** Success.

› **500** Internal server error.

GET /api/v1/assets

### Response samples

**200**

**500**

#### Content type

application/json; charset=utf-8

Copy    Expand all    Collapse all

```
[  
- {  
    "symbol": "BTC",  
    "displayName": "string",  
    "coingeckoId": "string",  
    + "tokens": [ ... ]  
}  
]
```

## Get collateral.

Get collateral parameters for assets.

## Responses

› **200** Success.

› **500** Internal server error.

GET /api/v1/collateral

## Response samples

**200**

**500**

### Content type

application/json; charset=utf-8

[Copy](#) [Expand all](#) [Collapse all](#)

```
[  
- {  
    "symbol": "string",  
    + "imfFunction": { ... },  
    + "mmfFunction": { ... },  
    + "haircutFunction": { ... }  
}  
]
```

---

## Borrow Lend Markets

Borrowing and lending.

Get borrow lend markets.

## Responses

› **200** Success.

› **500** Internal server error.

GET /api/v1/borrowLend/markets

## Response samples

**200**

**500**

**Content type**

application/json; charset=utf-8

Copy    Expand all    Collapse all

```
[  
- {  
    "state": "Open",  
    "assetMarkPrice": "string",  
    "borrowInterestRate": "string",  
    "borrowedQuantity": "string",  
    "fee": "string",  
    "lendInterestRate": "string",  
    "lentQuantity": "string",  
    "maxUtilization": "string",  
    "openBorrowLendLimit": "string",  
    "optimalUtilization": "string",  
    "symbol": "BTC",  
    "timestamp": "2019-08-24T14:15:22Z",  
    "throttleUtilizationThreshold": "string",  
    "throttleUtilizationBound": "string",  
    "throttleUpdateFraction": "string",  
    "utilization": "string",  
    "stepSize": "string"  
}  
]  
]
```

---

## Get borrow lend market history.

### QUERY PARAMETERS

---

**interval** required string (BorrowLendMarketHistoryInterval)  
 Enum:      
 Filter for an interval.

---

**symbol** string  
 Market symbol to query. If not set, all markets are returned.

---

## Responses

> **200 Success.**

> **500 Internal server error.**

GET /api/v1/borrowLend/markets/history

## Response samples

200

500

### Content type

application/json; charset=utf-8

[Copy](#) [Expand all](#) [Collapse all](#)

```
[  
- {  
    "borrowInterestRate": "string",  
    "borrowedQuantity": "string",  
    "lendInterestRate": "string",  
    "lentQuantity": "string",  
    "timestamp": "2019-08-24T14:15:22Z",  
    "utilization": "string"  
}  
]
```

## Markets

Public market data.

### Get markets.

Retrieves all the markets that are supported by the exchange.

## Responses

› **200 Success.**

> **500 Internal server error.**

GET /api/v1/markets

## Response samples

200

500

### Content type

application/json; charset=utf-8

[Copy](#) [Expand all](#) [Collapse all](#)

```
[  
  - {  
      "symbol": "string",  
      "baseSymbol": "BTC",  
      "quoteSymbol": "BTC",  
      "marketType": "SPOT",  
      + "filters": { ... },  
      + "imfFunction": { ... },  
      + "mmfFunction": { ... },  
      "fundingInterval": 0,  
      "fundingRateUpperBound": "string",  
      "fundingRateLowerBound": "string",  
      "openInterestLimit": "string",  
      "orderBookState": "Open",  
      "createdAt": "string",  
      "visible": true  
    }  
]
```

## Get market.

Retrieves a market supported by the exchange.

### QUERY PARAMETERS

symbol <b>required</b>	string
---------------------------	--------

## Responses

› **200** Success.

› **400** Bad request.

› **500** Internal server error.

GET /api/v1/market

## Response samples

200

400

500

### Content type

application/json; charset=utf-8

Copy    Expand all    Collapse all

```
{  
  "symbol": "string",  
  "baseSymbol": "BTC",  
  "quoteSymbol": "BTC",  
  "marketType": "SPOT",  
  - "filters": {  
      + "price": { ... },  
      + "quantity": { ... }  
    },  
  - "imffFunction": {  
      "type": "sqrt",  
      "base": "string",  
      "factor": "string"  
    },  
  - "mmffFunction": {  
      "type": "sqrt",  
      "base": "string",  
      "factor": "string"  
    },  
}
```

```
"fundingInterval": 0,  
"fundingRateUpperBound": "string",  
"fundingRateLowerBound": "string",  
"openInterestLimit": "string",  
"orderBookState": "Open",  
"createdAt": "string",  
"visible": true  
}
```

---

## Get ticker.

Retrieves summarised statistics for the last 24 hours for the given market symbol.

---

### QUERY PARAMETERS

**symbol**  
**required** string

**interval** string (TickerInterval)  
Enum: `"1d"` `"1w"`

---

## Responses

➤ **200** Success.

— **204** Not found.

➤ **400** Bad request.

➤ **500** Internal server error.

GET /api/v1/ticker

### Response samples

`200`

`400`

`500`

**Content type**

application/json; charset=utf-8

[Copy](#)

```
{  
    "symbol": "string",  
    "firstPrice": "string",  
    "lastPrice": "string",  
    "priceChange": "string",  
    "priceChangePercent": "string",  
    "high": "string",  
    "low": "string",  
    "volume": "string",  
    "quoteVolume": "string",  
    "trades": "string"  
}
```

## Get tickers.

Retrieves summarised statistics for the last 24 hours for all market symbols.

---

### QUERY PARAMETERS

---

interval	string (TickerInterval)
Enum:	<input type="button" value="1d"/> <input type="button" value="1w"/>

---

## Responses

› **200** Success.

› **500** Internal server error.

GET /api/v1/tickers

## Response samples

**200****500****Content type**

application/json; charset=utf-8

[Copy](#)   [Expand all](#)   [Collapse all](#)

```
[  
  - {  
      "symbol": "string",  
      "firstPrice": "string",  
      "lastPrice": "string",  
      "priceChange": "string",  
      "priceChangePercent": "string",  
      "high": "string",  
      "low": "string",  
      "volume": "string",  
      "quoteVolume": "string",  
      "trades": "string"  
    }  
]
```

## Get depth.

Retrieves the order book depth for a given market symbol.

---

### QUERY PARAMETERS

---

**symbol**  
**required** string

---

## Responses

› **200** Success.

› **400** Bad request.

› **500**

GET /api/v1/depth

## Response samples

**200**

**400**

**500**

### Content type

application/json; charset=utf-8

[Copy](#) [Expand all](#) [Collapse all](#)

```
{
  - "asks": [
    + [ ... ],
    + [ ... ]
  ],
  - "bids": [
    + [ ... ],
    + [ ... ]
  ],
  "lastUpdateId": "1684026955123",
  "timestamp": 1684026955123
}
```

## Get K-lines.

Get K-Lines for the given market symbol, optionally providing a `startTime` and `endTime`. If no `endTime` is provided, the current time will be used.

The `priceType` parameter can be used to specify the price type of the kline. If not provided, the default is `LastPrice`.

### QUERY PARAMETERS

<code>symbol</code> <b>required</b>	string Market symbol for the kline query, e.g. SOL_USDC.
<code>interval</code> <b>required</b>	string (KlineInterval) Enum: <code>"1m"</code> <code>"3m"</code> <code>"5m"</code> <code>"15m"</code> <code>"30m"</code> <code>"1h"</code> <code>"2h"</code> <code>"4h"</code> <code>"6h"</code> <code>"8h"</code> <code>"12h"</code> <code>"1d"</code> <code>"3d"</code> <code>"1w"</code> <code>"1month"</code>
<code>startTime</code>	integer <int64>

**required**

UTC timestamp in seconds.

endTime

integer &lt;int64&gt;

UTC timestamp in seconds. Set to the current time if not provided.

priceType

string (KlinePriceType)

Enum: "Last" "Index" "Mark"

The price type of the K-line.

## Responses

**> 200 Success.****> 400 Bad request.****> 500 Internal server error.**

GET /api/v1/klines

## Response samples

**200****400****500****Content type**

application/json; charset=utf-8

[Copy](#) [Expand all](#) [Collapse all](#)

```
[  
  - {  
      "start": "string",  
      "end": "string",  
      "open": "string",  
      "high": "string",  
      "low": "string",  
      "close": "string",  
      "volume": "string",  
      "quoteVolume": "string",  
      "trades": "string"  
    }  
]
```

---

## Get all mark prices.

Retrieves mark price, index price and the funding rate for the current interval for all symbols, or the symbol specified.

---

### QUERY PARAMETERS

---

symbol	string
--------	--------

---

## Responses

› **200** Success.

› **400** Bad request.

› **500** Internal server error.

GET /api/v1/markPrices

## Response samples

**200****400****500****Content type**

application/json; charset=utf-8

[Copy](#) [Expand all](#) [Collapse all](#)

```
[  
- {  
    "fundingRate": "string",  
    "indexPrice": "string",  
    "markPrice": "string",  
    "nextFundingTimestamp": 0,  
    "symbol": "string"  
}  
]
```

## Get open interest.

Retrieves the current open interest for the given market. If no market is provided, then all markets are returned.

---

### QUERY PARAMETERS

---

symbol	string
--------	--------

---

## Responses

› **200** Success.

› **400** Bad request.

› **500** Internal server error.

GET /api/v1/openInterest

## Response samples

**200****400****500**

### Content type

application/json; charset=utf-8

[Copy](#) [Expand all](#) [Collapse all](#)

```
[
  - {
    "symbol": "string",
    "openInterest": "string",
    "timestamp": 0
  }
]
```

## Get funding interval rates.

Funding interval rate history for futures.

### QUERY PARAMETERS

**symbol**  
required

string

Market symbol to query

**limit**

integer &lt;uint64&gt;

Maximum number to return. Default **100**, maximum **10000**.**offset**

integer &lt;uint64&gt;

Offset for pagination. Default **0**.

## Responses

> **200** Success.

> **400** Bad request.

> **500** Internal server error.

GET /api/v1/fundingRates

## Response samples

200

400

500

### Content type

application/json; charset=utf-8

[Copy](#) [Expand all](#) [Collapse all](#)

```
[  
- {  
    "symbol": "string",  
    "intervalEndTimestamp": "string",  
    "fundingRate": "string"  
}  
]
```

## System

Exchange system status.

## Status.

Get the system status, and the status message, if any.

## Responses

› **200 Success.**

```
GET /api/v1/status
```

## Response samples

200

### Content type

application/json; charset=utf-8

Copy

```
{  
  "status": "Ok",  
  "message": "string"  
}
```

## Ping.

Responds with pong.

## Responses

> 200

```
GET /api/v1/ping
```

## Get system time.

Retrieves the current system time.

## Responses

### > 200

GET /api/v1/time

## Get wallets.

Returns all configured blockchain wallets and their addresses.

## Responses

### > 200 Success.

GET /api/v1/wallets

## Response samples

200

### Content type

application/json; charset=utf-8

Copy    Expand all    Collapse all

```
[  
- {  
  "blockchain": "string",  
  "address": "string"  
}  
]
```

# Trades

Public trade data.

## Get recent trades.

Retrieve the most recent trades for a symbol. This is public data and is not specific to any account.

The maximum available recent trades is **1000**. If you need more than **1000** trades use the historical trades endpoint.

### QUERY PARAMETERS

---

**symbol**  
**required** string  
Market symbol to query fills for.

---

**limit** integer <uint16>  
Limit the number of fills returned. Default **100**, maximum **1000**.

---

## Responses

> **200** Success.

> **400** Bad request.

> **500** Internal Server Error.

GET /api/v1/trades

### Response samples

**200**

**400**

**500**

**Content type**

application/json; charset=utf-8

[Copy](#) [Expand all](#) [Collapse all](#)

```
[  
- {  
    "id": 0,  
    "price": "string",  
    "quantity": "string",  
    "quoteQuantity": "string",  
    "timestamp": 0,  
    "isBuyerMaker": true  
}  
]
```

## Get historical trades.

Retrieves all historical trades for the given symbol. This is public trade data and is not specific to any account.

---

### QUERY PARAMETERS

---

**symbol**  
**required**

string

**limit**

integer &lt;uint64&gt;

Limit the number of trades returned. Default **100**, maximum **1000**.

**offset**

integer &lt;uint64&gt;

Offset. Default **0**.

---

## Responses

› **200 Success.**

› **400 Bad request.**

› **500 Internal Server Error.**

GET /api/v1/trades/history

## Response samples

200

400

500

### Content type

application/json; charset=utf-8

[Copy](#) [Expand all](#) [Collapse all](#)

```
[  
- {  
    "id": 0,  
    "price": "string",  
    "quantity": "string",  
    "quoteQuantity": "string",  
    "timestamp": 0,  
    "isBuyerMaker": true  
}  
]
```

## Account

Account settings and limits.

### Get account.

**Instruction:** [accountQuery](#)

#### HEADER PARAMETERS

X-API-KEY  
**required**

string

API key

X-SIGNATURE  
**required**

string

Signature of the request

X-TIMESTAMP <b>required</b>	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default <b>5000</b> , maximum <b>60000</b> )

## Responses

› **200** Success.

› **400** Bad request.

› **500** Internal server error.

GET /api/v1/account

## Response samples

**200**

**400**

**500**

**Content type**

application/json; charset=utf-8

Copy

```
{
    "autoBorrowSettlements": true,
    "autoLend": true,
    "autoRealizePnl": true,
    "autoRepayBorrows": true,
    "borrowLimit": "string",
    "futuresMakerFee": "string",
    "futuresTakerFee": "string",
    "leverageLimit": "string",
    "limitOrders": 0,
    "liquidating": true,
    "positionLimit": "string",
    "spotMakerFee": "string",
    "spotTakerFee": "string",
```

```
"triggerOrders": 0  
}
```

## Update account.

Update account settings.

**Instruction:** `accountUpdate`

### HEADER PARAMETERS

X-API-KEY <b>required</b>	string API key
X-SIGNATURE <b>required</b>	string Signature of the request
X-TIMESTAMP <b>required</b>	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default <code>5000</code> , maximum <code>60000</code> )

REQUEST BODY SCHEMA: `application/json; charset=utf-8`  
**required**

<code>autoBorrowSettlements</code>	boolean If true, then tries to borrow during collateral reconciliation. Collateral reconciliation is a process in which the system reconciles the negative account debt or positive account equity.
<code>autoLend</code>	boolean Determines if the account should automatically lend.
<code>autoRepayBorrows</code>	boolean Determines if the account should automatically repay borrows with available balance.
<code>leverageLimit</code>	string <decimal> Determines the maximum leverage allowed for the main account or subaccount.

## Responses

— **200** Success.

> **400** Bad request.

> **401** Unauthorized.

> **500** Internal server error.

> **503** System under maintenance.

PATCH /api/v1/account

## Request samples

### Payload

Content type

application/json; charset=utf-8

Copy

```
{  
    "autoBorrowSettlements": true,  
    "autoLend": true,  
    "autoRepayBorrows": true,  
    "leverageLimit": "string"  
}
```

## Response samples

400

401

500

503

Content type

application/json; charset=utf-8

Copy

```
{
  "code": "ACCOUNT_LIQUIDATING",
  "message": "string"
}
```

## Convert a dust balance on an account.

Converts a dust balance to USDC. The balance (including lend) must be less than the minimum quantity tradable on the spot order book.

**Instruction:** `convertDust`

### HEADER PARAMETERS

X-API-KEY <b>required</b>	string API key
X-SIGNATURE <b>required</b>	string Signature of the request
X-TIMESTAMP <b>required</b>	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default <code>5000</code> , maximum <code>60000</code> )

### REQUEST BODY SCHEMA:

**required**

symbol	string
	Enum: <code>"BTC"</code> <code>"ETH"</code> <code>"SOL"</code> <code>"USDC"</code> <code>"USDT"</code> <code>"PYTH"</code> <code>"JTO"</code> <code>"BONK"</code> <code>"HNT"</code> <code>"MOBILE"</code> <code>"WIF"</code> <code>"JUP"</code> <code>"RENDER"</code> <code>"WEN"</code> <code>"W"</code> <code>"TNSR"</code> <code>"PRCL"</code> <code>"SHARK"</code> <code>"KMNO"</code> <code>"MEW"</code> <code>"BOME"</code> <code>"RAY"</code> <code>"HONEY"</code> <code>"SHFL"</code> <code>"BODEN"</code> <code>"IO"</code> <code>"DRIFT"</code> <code>"PEPE"</code> <code>"SHIB"</code> <code>"LINK"</code> <code>"UNI"</code> <code>"ONDO"</code> <code>"FTM"</code> <code>"MATIC"</code> <code>"STRK"</code> <code>"BLUR"</code> <code>"WLD"</code> <code>"GALA"</code> <code>"NYAN"</code> <code>"HLG"</code> <code>"MON"</code> <code>"ZKJ"</code> <code>"MANEKI"</code> <code>"HABIBI"</code> <code>"UNA"</code> <code>"ZRO"</code> <code>"ZEX"</code> <code>"AAVE"</code> <code>"LDO"</code> <code>"MOTHER"</code> <code>"CLOUD"</code> <code>"MAX"</code> <code>"POL"</code> <code>"TRUMPWIN"</code> <code>"HARRISWIN"</code> <code>"MOODENG"</code> <code>"DBR"</code> <code>"GOAT"</code> <code>"ACT"</code> <code>"DOGE"</code> <code>"BCH"</code> <code>"LTC"</code> <code>"APE"</code> <code>"ENA"</code> <code>"ME"</code> <code>"EIGEN"</code> <code>"CHILLGUY"</code> <code>"PENGU"</code> <code>"EUR"</code> <code>"SONIC"</code> <code>"J"</code> <code>"TRUMP"</code> <code>"MELANIA"</code> <code>"ANIME"</code> <code>"XRP"</code> <code>"SUI"</code> <code>"VINE"</code> <code>"ADA"</code> <code>"MOVE"</code> <code>"BERA"</code> <code>"IP"</code> <code>"HYPE"</code> <code>"BNB"</code> <code>"KAITO"</code> <code>"kPEPE"</code> <code>"kBONK"</code> <code>"kSHIB"</code> <code>"AVAX"</code> <code>"S"</code> <code>"POINTS"</code> <code>"ROAM"</code> <code>"AI16Z"</code>

"LAYER"	"FARTCOIN"	"NEAR"	"PNUT"	"ARB"	"DOT"	"APT"
"OP"	"PYUSD"	"HUMA"	"WAL"	"DEEP"	"CETUS"	"SEND"
"BLUE"	"NS"	"HAEDAL"	"JPY"	"TAO"	"VIRTUAL"	"TIA"
"TRX"	"FRAG"	"PUMP"	"WCT"	"ES"	"SEI"	"CRV"
"HBAR"	"XLM"	"ZORA"	"WLFI"	"BPEUR"	"SWTCH"	"LINEA"
"XPL"	"BARD"	"FLOCK"	"AVNT"	"PENDLE"	"AERO"	"ASTER"
"GLXY"	"OG"	"2Z"	"FORD"	"ZEUS"	"APEX"	"EDEN"
"ORDER"	"MNT"	"ZEC"	"PAXG"	"MORPHO"	"ATH"	"KGEN"
"XAUT"	"FOGO"	"SPX"	"ETHFI"	"APR"	"PIPE"	"MET"

The asset symbol to convert dust for. If omitted, all dust balances will be converted.

---

## Responses

— **200** Success.

> **400** Bad request.

> **500** Internal server error.

> **503** System under maintenance.

POST /api/v1/account/convertDust

## Request samples

**Payload**

**Content type**

application/json; charset=utf-8

Copy

```
{
  "symbol": "BTC"
}
```

## Response samples

**400**

**500**

**503**

**Content type**

application/json; charset=utf-8

{

```
"code": "ACCOUNT_LIQUIDATING",
"message": "string"
```

}

[Copy](#)

## Get max borrow quantity.

Retrieves the maximum quantity an account can borrow for a given asset based on the accounts existing exposure and margin requirements

**Instruction:** [maxBorrowQuantity](#)

---

### QUERY PARAMETERS

<b>symbol</b> <b>required</b>	string The asset to borrow.
----------------------------------	--------------------------------

---

---

### HEADER PARAMETERS

<b>X-API-KEY</b>	string API key
<b>X-SIGNATURE</b>	string Signature of the request
<b>X-TIMESTAMP</b>	integer <int64> Timestamp of the request in milliseconds
<b>X-WINDOW</b>	integer <uint64> Time the request is valid for in milliseconds (default <a href="#">5000</a> , maximum <a href="#">60000</a> )

---

## Responses

> **200 Success.**

> **400** Bad request.

> **500** Internal server error.

> **503** Service unavailable.

GET /api/v1/account/limits/borrow

## Response samples

200

400

500

503

### Content type

application/json; charset=utf-8

Copy

```
{  
    "maxBorrowQuantity": "string",  
    "symbol": "string"  
}
```

## Get max order quantity.

Retrieves the maximum quantity an account can trade for a given symbol based on the account's balances, existing exposure and margin requirements.

**Instruction:** `maxOrderQuantity`

### QUERY PARAMETERS

**symbol** string  
**required** The market symbol to trade.

**side** string (Side)  
**required** Enum: `"Bid"` `"Ask"`  
The side of the order.

price	string <decimal> The limit price of the order. Not included for market orders.
reduceOnly	boolean Whether the order is reduce only.
autoBorrow	boolean Whether the order uses auto borrow.
autoBorrowRepay	boolean Whether the order uses auto borrow repay.
autoLendRedeem	boolean Whether the order uses auto lend redeem.

## HEADER PARAMETERS

X-API-KEY	string API key
X-SIGNATURE	string Signature of the request
X-TIMESTAMP	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default <b>5000</b> , maximum <b>60000</b> )

## Responses

› **200** Success.

› **400** Bad request.

› **500** Internal server error.

GET /api/v1/account/limits/order

## Response samples

**200****400****500****Content type**

application/json; charset=utf-8

[Copy](#)

```
{
    "autoBorrow": true,
    "autoBorrowRepay": true,
    "autoLendRedeem": true,
    "maxOrderQuantity": "string",
    "price": "string",
    "side": "string",
    "symbol": "string",
    "reduceOnly": true
}
```

## Get max withdrawal quantity.

Retrieves the maximum quantity an account can withdraw for a given asset based on the accounts existing exposure and margin requirements. The response will include the maximum quantity that can be withdrawn and whether the withdrawal is with auto borrow or auto lend redeem enabled.

**Instruction:** `maxWithdrawalQuantity`

---

### QUERY PARAMETERS

<code>symbol</code> <b>required</b>	string The asset to withdraw.
<code>autoBorrow</code>	boolean Whether the withdrawal is with auto borrow.
<code>autoLendRedeem</code>	boolean Whether the withdrawal is with auto lend redeem.

---

### HEADER PARAMETERS

<code>X-API-KEY</code>	string API key
------------------------	-------------------

X-SIGNATURE	string Signature of the request
X-TIMESTAMP	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default <b>5000</b> , maximum <b>60000</b> )

## Responses

> **200** Success.

> **400** Bad request.

> **500** Internal server error.

GET /api/v1/account/limits/withdrawal

## Response samples

**200**

**400**

**500**

### Content type

application/json; charset=utf-8

Copy

```
{
  "autoBorrow": true,
  "autoLendRedeem": true,
  "maxWithdrawQuantity": "string",
  "symbol": "string"
}
```

## Borrow Lend

Borrowing and lending.

## Get borrow lend positions.

Retrieves all the open borrow lending positions for the account.

**Instruction:** `borrowLendPositionQuery`

### HEADER PARAMETERS

---

X-API-KEY

string

API key

---

X-SIGNATURE

string

Signature of the request

---

X-TIMESTAMP

integer <int64>

Timestamp of the request in milliseconds

---

X-WINDOW

integer <uint64>

Time the request is valid for in milliseconds (default `5000`, maximum `60000`)

---

## Responses

> **200** Success.

> **400** Bad request.

> **500** Internal server error.

GET /api/v1/borrowLend/positions

### Response samples

`200`

`400`

`500`

**Content type**

application/json; charset=utf-8

[Copy](#)
[Expand all](#)
[Collapse all](#)

```
[
  - {
    "cumulativeInterest": "string",
    "id": "string",
    "imf": "string",
    + "imfFunction": { ... },
    "netQuantity": "string",
    "markPrice": "string",
    "mmf": "string",
    + "mmfFunction": { ... },
    "netExposureQuantity": "string",
    "netExposureNotional": "string",
    "symbol": "BTC"
  }
]
```

## Execute borrow lend.

**Instruction:** [borrowLendExecute](#)

### HEADER PARAMETERS

X-API-KEY <b>required</b>	string API key
X-SIGNATURE <b>required</b>	string Signature of the request
X-TIMESTAMP <b>required</b>	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default <b>5000</b> , maximum <b>60000</b> )

REQUEST BODY SCHEMA: application/json; charset=utf-8  
**required**

quantity <b>required</b>	string <decimal> The quantity of the asset to repay.
side <b>required</b>	string Enum: "Borrow" "Lend" Side of the borrow lend.
symbol <b>required</b>	string Enum: "BTC" "ETH" "SOL" "USDC" "USDT" "PYTH" "JTO" "BONK" "HNT" "MOBILE" "WIF" "JUP" "RENDER" "WEN" "W" "TNSR" "PRCL" "SHARK" "KMNO" "MEW" "BOME" "RAY" "HONEY" "SHFL" "BODEN" "IO" "DRIFT" "PEPE" "SHIB" "LINK" "UNI" "ONDO" "FTM" "MATIC" "STRK" "BLUR" "WLD" "GALA" "NYAN" "HLG" "MON" "ZKJ" "MANEKI" "HABIBI" "UNA" "ZRO" "ZEX" "AAVE" "LDO" "MOTHER" "CLOUD" "MAX" "POL" "TRUMPWIN" "HARRISWIN" "MOODENG" "DBR" "GOAT" "ACT" "DOGE" "BCH" "LTC" "APE" "ENA" "ME" "EIGEN" "CHILLGUY" "PENGU" "EUR" "SONIC" "J" "TRUMP" "MELANIA" "ANIME" "XRP" "SUI" "VINE" "ADA" "MOVE" "BERA" "IP" "HYPE" "BNB" "KAITO" "kPEPE" "kBONK" "kSHIB" "AVAX" "S" "POINTS" "ROAM" "AI16Z" "LAYER" "FARTCOIN" "NEAR" "PNUT" "ARB" "DOT" "APT" "OP" "PYUSD" "HUMA" "WAL" "DEEP" "CETUS" "SEND" "BLUE" "NS" "HAEDAL" "JPY" "TAO" "VIRTUAL" "TIA" "TRX" "FRAG" "PUMP" "WCT" "ES" "SEI" "CRV" "TON" "HBAR" "XLM" "ZORA" "WLF1" "BPEUR" "SWTCH" "LINEA" "XPL" "BARD" "FLOCK" "AVNT" "PENDLE" "AERO" "ASTER" "GLXY" "OG" "2Z" "FORD" "ZEUS" "APEX" "EDEN" "FF" "ORDER" "MNT" "ZEC" "PAXG" "MORPHO" "ATH" "KGEN" "XAUT" "FOGO" "SPX" "ETHFI" "APR" "PIPE" "MET"
	The asset to repay.

## Responses

— **200 Success.**

> **400 Bad request.**

> **500 Internal server error.**

> **503 System under maintenance.**

POST /api/v1/borrowLend

## Request samples

### Payload

#### Content type

application/json; charset=utf-8

Copy

```
{  
    "quantity": "string",  
    "side": "Borrow",  
    "symbol": "BTC"  
}
```

## Response samples

400

500

503

#### Content type

application/json; charset=utf-8

Copy

```
{  
    "code": "ACCOUNT_LIQUIDATING",  
    "message": "string"  
}
```

Get an estimated liquidation price for potential borrow lend position.

Retrieves the estimated liquidation price for a potential borrow lend position.

## QUERY PARAMETERS

subaccountId      integer <uint16>  
Optional subaccount.

borrow <b>required</b>	string	Standard base64 encoded json of [ <a href="#">BorrowLendExecutePayload</a> ]
---------------------------	--------	--

## Responses

- > **200** Success.
- > **400** Bad request.
- > **401** Unauthorized.
- > **500** Internal server error.

GET /api/v1/borrowLend/position/liquidationPrice

## Response samples

[200](#) [400](#) [401](#) [500](#)

### Content type

application/json; charset=utf-8

Copy

```
{  
    "liquidationPrice": "string",  
    "markPrice": "string"  
}
```

## Capital

Capital management.

## Get balances.

Retrieves account balances and the state of the balances (locked or available).

Locked assets are those that are currently in an open order.

**Instruction:** `balanceQuery`

### HEADER PARAMETERS

---

X-API-KEY	string API key
X-SIGNATURE	string Signature of the request
X-TIMESTAMP	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default <code>5000</code> , maximum <code>60000</code> )

---

## Responses

› **200** Success.

› **400** Bad request.

› **500** Internal server error.

GET /api/v1/capital

### Response samples

`200`

`400`

`500`

**Content type**

application/json; charset=utf-8

```
{
  - "property1": {
    "available": "string",
    "locked": "string",
    "staked": "string"
  },
  - "property2": {
    "available": "string",
    "locked": "string",
    "staked": "string"
  }
}
```

## Get collateral.

Retrieves collateral information for an account.

**Instruction:** `collateralQuery`

### QUERY PARAMETERS

`subaccountId`    integer <uint16>

### HEADER PARAMETERS

`X-API-KEY`    string  
API key

`X-SIGNATURE`    string  
Signature of the request

`X-TIMESTAMP`    integer <int64>  
Timestamp of the request in milliseconds

`X-WINDOW`    integer <uint64>  
Time the request is valid for in milliseconds (default `5000`, maximum `60000`)

## Responses

> **200** Success.

> **400** Bad request.

> **500** Internal server error.

GET /api/v1/capital/collateral

### Response samples

**200**

**400**

**500**

#### Content type

application/json; charset=utf-8

[Copy](#) [Expand all](#) [Collapse all](#)

```
{  
    "assetsValue": "string",  
    "borrowLiability": "string",  
    - "collateral": [  
        + { ... }  
    ],  
    "imf": "string",  
    "unsettledEquity": "string",  
    "liabilitiesValue": "string",  
    "marginFraction": "string",  
    "mmf": "string",  
    "netEquity": "string",  
    "netEquityAvailable": "string",  
    "netEquityLocked": "string",  
    "netExposureFutures": "string",  
    "pnlUnrealized": "string"  
}
```

## Get deposits.

Retrieves deposit history.

Instruction: `depositQueryAll`

## QUERY PARAMETERS

---

from	integer <int64> Filter to minimum time (milliseconds).
to	integer <int64> Filter to maximum time (milliseconds).
limit	integer <uint64> Maximum number to return. Default <code>100</code> , maximum <code>1000</code> .
offset	integer <uint64> Offset. Default <code>0</code> .

## HEADER PARAMETERS

---

X-API-KEY	string API key
X-SIGNATURE	string Signature of the request
X-TIMESTAMP	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default <code>5000</code> , maximum <code>60000</code> )

## Responses

> **200** Success.

> **400** Bad request.

> **401** Unauthorized.

> **500** Internal server error.

GET /wapi/v1/capital/deposits

## Response samples

[200](#)
[400](#)
[401](#)
[500](#)

### Content type

application/json; charset=utf-8

[Copy](#) [Expand all](#) [Collapse all](#)

```
[  
  - {  
      "id": 0,  
      "toAddress": "string",  
      "fromAddress": "string",  
      "source": "administrator",  
      "status": "cancelled",  
      "transactionHash": "string",  
      "symbol": "BTC",  
      "quantity": "string",  
      "createdAt": "string",  
      "fiatAmount": 0.1,  
      "fiatCurrency": "AED",  
      "institutionBic": "string",  
      "platformMemo": "string"  
    }  
]
```

## Get deposit address.

Retrieves the user specific deposit address if the user were to deposit on the specified blockchain.

**Instruction:** [depositAddressQuery](#)

### QUERY PARAMETERS

<b>blockchain</b>	string (Blockchain)
<b>required</b>	Enum: <a href="#">"OG"</a> <a href="#">"Aptos"</a> <a href="#">"Arbitrum"</a> <a href="#">"Avalanche"</a> <a href="#">"Base"</a> <a href="#">"Berachain"</a> <a href="#">"Bitcoin"</a> <a href="#">"BitcoinCash"</a> <a href="#">"Bsc"</a> <a href="#">"Cardano"</a> <a href="#">"Dogecoin"</a> <a href="#">"Eclipse"</a> <a href="#">"EqualsMoney"</a> <a href="#">"Ethereum"</a>

"HyperEVM" "Hyperliquid" "Linea" "Litecoin" "Optimism"  
 "Plasma" "Polygon" "Sei" "Sui" "Solana" "Story"  
 "Tron" "XRP" "Zcash"

Blockchain symbol to get a deposit address for.

## HEADER PARAMETERS

### X-API-KEY

string

API key

### X-SIGNATURE

string

Signature of the request

### X-TIMESTAMP

integer <int64>

Timestamp of the request in milliseconds

### X-WINDOW

integer <uint64>

Time the request is valid for in milliseconds (default **5000**, maximum **60000**)

## Responses

> **200** Success.

> **400** Bad request.

> **401** Unauthorized.

> **409** Conflict

> **500** Internal server error.

GET /wapi/v1/capital/deposit/address

## Response samples

**200**

**400**

**401**

**409**

**500**

**Content type**

application/json; charset=utf-8

[Copy](#)

```
{  
    "address": "string"  
}
```

## Get withdrawals.

Retrieves withdrawal history.

**Instruction:** [withdrawalQueryAll](#)

### QUERY PARAMETERS

from	integer <int64> Filter to minimum time (milliseconds).
to	integer <int64> Filter to maximum time (milliseconds).
limit	integer <uint64> Maximum number to return. Default <a href="#">100</a> , maximum <a href="#">1000</a> .
offset	integer <uint64> Offset. Default <a href="#">0</a> .

### HEADER PARAMETERS

X-API-KEY	string API key
X-SIGNATURE	string Signature of the request
X-TIMESTAMP	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default <a href="#">5000</a> , maximum <a href="#">60000</a> )

## Responses

> **200** Success.

> **400** Bad request.

> **401** Unauthorized.

> **500** Internal Server Error.

GET /wapi/v1/capital/withdrawals

## Response samples

**200**

**400**

**401**

**500**

### Content type

application/json; charset=utf-8

[Copy](#) [Expand all](#) [Collapse all](#)

```
[  
- {  
    "id": 0,  
    "blockchain": "0G",  
    "clientId": "string",  
    "identifier": "string",  
    "quantity": "string",  
    "fee": "string",  
    "fiatFee": "string",  
    "fiatState": "initialized",  
    "fiatSymbol": "AED",  
    "providerId": "string",  
    "symbol": "BTC",  
    "status": "confirmed",  
    "subaccountId": 0,  
    "toAddress": "string",  
    "transactionHash": "string",  
    "createdAt": "string",  
    "isInternal": true,  
    "bankName": "string",  
    "bankIdentifier": "string",  
    "accountIdentifier": "string",  
    "triggerAt": "string"  
}
```

]

## Request withdrawal.

Requests a withdrawal from the exchange.

The `twoFactorToken` field is required if the withdrawal address is not an address that is configured in the address book to not require 2FA. These addresses can be configured [here](#).

**Instruction:** `withdraw`

### HEADER PARAMETERS

X-API-KEY <b>required</b>	string API key
X-TIMESTAMP <b>required</b>	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default <code>5000</code> , maximum <code>60000</code> )
X-SIGNATURE <b>required</b>	string Signature of the request

REQUEST BODY SCHEMA: `application/json; charset=utf-8`  
**required**

address <b>required</b>	string Address to withdraw to.
blockchain <b>required</b>	string Enum: <code>"0G"</code> <code>"Aptos"</code> <code>"Arbitrum"</code> <code>"Avalanche"</code> <code>"Base"</code> <code>"Berachain"</code> <code>"Bitcoin"</code> <code>"BitcoinCash"</code> <code>"Bsc"</code> <code>"Cardano"</code> <code>"Dogecoin"</code> <code>"Eclipse"</code> <code>"EqualsMoney"</code> <code>"Ethereum"</code> <code>"HyperEVM"</code> <code>"Hyperliquid"</code> <code>"Linea"</code> <code>"Litecoin"</code> <code>"Optimism"</code> <code>"Plasma"</code> <code>"Polygon"</code> <code>"Sei"</code> <code>"Sui"</code> <code>"Solana"</code> <code>"Story"</code> <code>"Tron"</code> <code>"XRP"</code> <code>"Zcash"</code> Blockchain to withdraw on.
clientId	string   <code>&lt;= 255 characters</code> Custom client id.
quantity	string <decimal>

**required**

Quantity to withdraw.

symbol

**required**

string

Enum: "BTC" "ETH" "SOL" "USDC" "USDT" "PYTH" "JTO"  
 "BONK" "HNT" "MOBILE" "WIF" "JUP" "RENDER" "WEN" "W"  
 "TNSR" "PRCL" "SHARK" "KMNO" "MEW" "BOME" "RAY"  
 "HONEY" "SHFL" "BODEN" "IO" "DRIFT" "PEPE" "SHIB"  
 "LINK" "UNI" "ONDO" "FTM" "MATIC" "STRK" "BLUR"  
 "WLD" "GALA" "NYAN" "HLG" "MON" "ZKJ" "MANEKI"  
 "HABIBI" "UNA" "ZRO" "ZEX" "AAVE" "LDO" "MOTHER"  
 "CLOUD" "MAX" "POL" "TRUMPWIN" "HARRISWIN" "MOODENG"  
 "DBR" "GOAT" "ACT" "DOGE" "BCH" "LTC" "APE" "ENA"  
 "ME" "EIGEN" "CHILLGUY" "PENGU" "EUR" "SONIC" "J"  
 "TRUMP" "MELANIA" "ANIME" "XRP" "SUI" "VINE" "ADA"  
 "MOVE" "BERA" "IP" "HYPE" "BNB" "KAITO" "KPEPE"  
 "kBONK" "kSHIB" "AVAX" "S" "POINTS" "ROAM" "AI16Z"  
 "LAYER" "FARTCOIN" "NEAR" "PNUT" "ARB" "DOT" "APT"  
 "OP" "PYUSD" "HUMA" "WAL" "DEEP" "CETUS" "SEND"  
 "BLUE" "NS" "HAEDAL" "JPY" "TAO" "VIRTUAL" "TIA"  
 "TRX" "FRAG" "PUMP" "WCT" "ES" "SEI" "CRV" "TON"  
 "HBAR" "XLM" "ZORA" "WLF1" "BPEUR" "SWTCH" "LINEA"  
 "XPL" "BARD" "FLOCK" "AVNT" "PENDLE" "AERO" "ASTER"  
 "GLXY" "OG" "2Z" "FORD" "ZEUS" "APEX" "EDEN" "FF"  
 "ORDER" "MNT" "ZEC" "PAXG" "MORPHO" "ATH" "KGEN"  
 "XAUT" "FOGO" "SPX" "ETHFI" "APR" "PIPE" "MET"

Symbol of the asset to withdraw.

twoFactorToken

string

Issued two factor token.

autoBorrow

boolean

Auto borrow to withdraw if required.

autoLendRedeem

boolean

Auto redeem a lend if required.

## Responses

> **200 Success.**

> **400 Bad request.**

> **401 Unauthorized.**

> **403 Forbidden.**

## › 429 Too many requests.

## › 500 Internal server error.

› 503 System under maintenance.

POST /wapi/v1/capital/withdrawals

## Request samples

## Payload

## Content type

application/json; charset=utf-8

Copy

```
{  
    "address": "string",  
    "blockchain": "0G",  
    "clientId": "string",  
    "quantity": "string",  
    "symbol": "BTC",  
    "twoFactorToken": "string",  
    "autoBorrow": true,  
    "autoLendRedeem": true  
}
```

## Response samples

300

400

401

403

429

500

503

## Content type

application/json; charset=utf-8

Copy

```
{  
  "id": 0,  
  "blockchain": "0G",  
  "clientId": "string",  
  "identifier": "string",  
  "quantity": "string",
```

```
"fee": "string",
"fiatFee": "string",
"fiatState": "initialized",
"fiatSymbol": "AED",
"providerId": "string",
"symbol": "BTC",
"status": "confirmed",
"subaccountId": 0,
"toAddress": "string",
"transactionHash": "string",
"createdAt": "string",
"isInternal": true,
"bankName": "string",
"bankIdentifier": "string",
"accountIdentifier": "string",
"triggerAt": "string"
}
```

---

## Futures

Futures data.

### Get open positions.

Retrieves account position summary.

**Instruction:** `positionQuery`

---

#### QUERY PARAMETERS

<code>symbol</code>	string
Filter for a single position by symbol.	

---

#### HEADER PARAMETERS

---

<code>X-API-KEY</code>	string
API key	
<code>X-SIGNATURE</code>	string

---

### Signature of the request

X-TIMESTAMP

integer <int64>

Timestamp of the request in milliseconds

X-WINDOW

integer <uint64>

Time the request is valid for in milliseconds (default **5000**, maximum **60000**)

## Responses

> **200** Success.

> **400** Bad request.

> **401** Unauthorized.

> **404** Position not found.

> **500** Internal server error.

GET /api/v1/position

## Response samples

**200**

**400**

**401**

**404**

**500**

**Content type**

application/json; charset=utf-8

Copy    Expand all    Collapse all

```
[
  - {
    "breakEvenPrice": "string",
    "entryPrice": "string",
    "estLiquidationPrice": "string",
    "imf": "string",
    + "imfFunction": { ... },
    "markPrice": "string",
    "mmf": "string",
    + "mmfFunction": { ... },
    "netCost": "string",
    "netQuantity": "string",
    "netExposureQuantity": "string",
    "netExposureNotional": "string",
    "pnlRealized": "string",
    "pnlUnrealized": "string",
    "cumulativeFundingPayment": "string",
    "subaccountId": 0,
    "symbol": "string",
    "userId": 0,
    "positionId": "string",
    "cumulativeInterest": "string"
  }
]
```

## History

Historical account data.

### Get borrow history.

History of borrow and lend operations for the account.

**Instruction:** [borrowHistoryQueryAll](#)

#### QUERY PARAMETERS

type	string (BorrowLendEventType)
	Enum: <a href="#">"Borrow"</a> <a href="#">"BorrowRepay"</a> <a href="#">"Lend"</a> <a href="#">"LendRedeem"</a>

Filter to history for either borrows or lends.

---

<code>sources</code>	string
	Filter to return history for a particular source. Can be a single source, or multiple sources separated by commas.
<code>positionId</code>	string
	Filter to return history for a borrow lend position.
<code>symbol</code>	string
	Filter to the given symbol.
<code>limit</code>	integer <uint64>
	Maximum number to return. Default <code>100</code> , maximum <code>1000</code> .
<code>offset</code>	integer <uint64>
	Offset for pagination. Default <code>0</code> .
<code>sortDirection</code>	string (SortDirection)
	Enum: <code>"Asc"</code> <code>"Desc"</code>
	Sort direction.

---

## HEADER PARAMETERS

---

<code>X-API-KEY</code>	string
	API key
<code>X-SIGNATURE</code>	string
	Signature of the request
<code>X-TIMESTAMP</code>	integer <int64>
	Timestamp of the request in milliseconds
<code>X-WINDOW</code>	integer <uint64>
	Time the request is valid for in milliseconds (default <code>5000</code> , maximum <code>60000</code> )

---

## Responses

› **200 Success.**

› **400 Bad request.**

› **401 Unauthorized.**

> **500 Internal server error.**

GET /wapi/v1/history/borrowLend

## Response samples

200

400

401

500

### Content type

application/json; charset=utf-8

[Copy](#) [Expand all](#) [Collapse all](#)

```
[  
  - {  
      "eventType": "Borrow",  
      "positionId": "string",  
      "positionQuantity": "string",  
      "quantity": "string",  
      "source": "AdlProvider",  
      "symbol": "string",  
      "timestamp": "string",  
      "spotMarginOrderId": "string"  
    }  
]
```

## Get interest history.

History of the interest payments for borrows and lends for the account.

**Instruction:** [interestHistoryQueryAll](#)

### QUERY PARAMETERS

asset

string

Asset to query. If not set, all assets are returned.

symbol

string

Market symbol to query. If not set, all markets are returned. If a futures symbol is supplied, then interest payments on unrealized pnl will be returned.

Spot market symbols refer to interest payments on regular borrow lend positions.

---

positionId	string	Filter to return history for a borrow lend position.
limit	integer <uint64>	Maximum number to return. Default <code>100</code> , maximum <code>1000</code> .
offset	integer <uint64>	Offset for pagination. Default <code>0</code> .
source	string (InterestPaymentSource)	Enum: <code>"UnrealizedPnl"</code> <code>"BorrowLend"</code> Filter to return interest payments of a particular source. Borrow interest payments through two mechanisms: borrow lend positions; interest paid on unrealized pnl.
sortDirection	string (SortDirection)	Enum: <code>"Asc"</code> <code>"Desc"</code> Sort direction.

---

## HEADER PARAMETERS

---

X-API-KEY	string	API key
X-SIGNATURE	string	Signature of the request
X-TIMESTAMP	integer <int64>	Timestamp of the request in milliseconds
X-WINDOW	integer <uint64>	Time the request is valid for in milliseconds (default <code>5000</code> , maximum <code>60000</code> )

---

## Responses

› **200 Success.**

› **400 Bad request.**

› **401 Unauthorized.**

> **500 Internal server error.**

GET /wapi/v1/history/interest

## Response samples

200

400

401

500

### Content type

application/json; charset=utf-8

[Copy](#) [Expand all](#) [Collapse all](#)

```
[  
  - {  
      "paymentType": "EntryFee",  
      "interestRate": "string",  
      "interval": 0,  
      "marketSymbol": "string",  
      "positionId": "string",  
      "quantity": "string",  
      "symbol": "BTC",  
      "timestamp": "string"  
    }  
]
```

## Get borrow position history.

History of borrow and lend positions for the account.

**Instruction:** [borrowPositionHistoryQueryAll](#)

### QUERY PARAMETERS

symbol

string

Filter to the given symbol.

side	string (BorrowLendSide) Enum: "Borrow" "Lend" Return only borrows or only lends.
state	string (BorrowLendPositionState) Enum: "Open" "Closed" Return only open positions or closed positions.
limit	integer <uint64> Maximum number to return. Default 100, maximum 1000.
offset	integer <uint64> Offset for pagination. Default 0.
sortDirection	string (SortDirection) Enum: "Asc" "Desc" Sort direction.

## HEADER PARAMETERS

X-API-KEY	string API key
X-SIGNATURE	string Signature of the request
X-TIMESTAMP	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default 5000, maximum 60000)

## Responses

- › **200** Success.
- › **400** Bad request.
- › **401** Unauthorized.
- › **500** Internal server error.

GET /wapi/v1/history/borrowLend/positions

## Response samples

**200**

**400**

**401**

**500**

### Content type

application/json; charset=utf-8

[Copy](#) [Expand all](#) [Collapse all](#)

```
[  
  - {  
      "positionId": "string",  
      "quantity": "string",  
      "symbol": "string",  
      "source": "AdlProvider",  
      "cumulativeInterest": "string",  
      "avgInterestRate": "string",  
      "side": "Borrow",  
      "createdAt": "string"  
    }  
]
```

## Get dust conversion history.

Retrieves the dust conversion history for the user.

**Instruction:** [dustHistoryQueryAll](#)

### QUERY PARAMETERS

**id** integer <int64>  
Filter to a given dust conversion id.

**symbol** string  
Filter to the given symbol.

**limit** integer <uint64>  
Maximum number to return. Default [100](#), maximum [1000](#).

**offset** integer <uint64>  
Offset. Default **0**.

**sortDirection** string (SortDirection)  
Enum: **"Asc"** **"Desc"**  
Sort direction.

## HEADER PARAMETERS

**X-API-KEY**

string

API key

**X-SIGNATURE**

string

Signature of the request

**X-TIMESTAMP**

integer <int64>

Timestamp of the request in milliseconds

**X-WINDOW**

integer <uint64>

Time the request is valid for in milliseconds (default **5000**, maximum **60000**)

## Responses

> **200** Success.

> **400** Bad request.

> **401** Unauthorized.

> **500** Internal server error.

GET /wapi/v1/history/dust

## Response samples

**200**

**400**

**401**

**500**

### Content type

application/json; charset=utf-8

```
[
  - {
    "id": 0,
    "quantity": "string",
    "symbol": "string",
    "usdcReceived": "string",
    "timestamp": "string"
  }
]
```

---

## Get fill history.

Retrieves historical fills, with optional filtering for a specific order or symbol.

**Instruction:** [fillHistoryQueryAll](#)

### QUERY PARAMETERS

orderId	string	Filter to the given order.
strategyId	string	Filter to the given strategy.
from	integer <int64>	Filter to minimum time (milliseconds).
to	integer <int64>	Filter to maximum time (milliseconds).
symbol	string	Filter to the given symbol.
limit	integer <uint64>	Maximum number to return. Default <a href="#">100</a> , maximum <a href="#">1000</a> .
offset	integer <uint64>	Offset. Default <a href="#">0</a> .
fillType	string (FillType)	Enum: <a href="#">"User"</a> <a href="#">"BookLiquidation"</a> <a href="#">"Adl"</a> <a href="#">"Backstop"</a> <a href="#">"Liquidation"</a> <a href="#">"AllLiquidation"</a> <a href="#">"CollateralConversion"</a> <a href="#">"CollateralConversionAndSpotLiquidation"</a>

Filter to return fills for different fill types

---

<b>marketType</b>	Array of strings (MarketType) Items Enum: "SPOT" "PERP" "IPERP" "DATED" "PREDICTION" "RFQ"
	Market type.

---

<b>sortDirection</b>	string (SortDirection) Enum: "Asc" "Desc"
	Sort direction.

---

#### HEADER PARAMETERS

---

<b>X-API-KEY</b>	string API key
<b>X-SIGNATURE</b>	string Signature of the request
<b>X-TIMESTAMP</b>	integer <int64> Timestamp of the request in milliseconds
<b>X-WINDOW</b>	integer <uint64> Time the request is valid for in milliseconds (default 5000, maximum 60000)

---

## Responses

› **200** Success.

› **400** Bad request.

› **401** Unauthorized.

› **500** Internal server error.

GET /wapi/v1/history/fills

## Response samples

**200**

**400**

**401**

**500**

**Content type**

application/json; charset=utf-8

Copy    Expand all    Collapse all

```
[
  - {
    "clientId": "string",
    "fee": "string",
    "feeSymbol": "string",
    "isMaker": true,
    "orderId": "string",
    "price": "string",
    "quantity": "string",
    "side": "Bid",
    "symbol": "string",
    "systemOrderType": "CollateralConversion",
    "timestamp": "string",
    "tradeId": 0
  }
]
```

---

## Get funding payments.

Users funding payment history for futures.

**Instruction:** [fundingHistoryQueryAll](#)

---

### QUERY PARAMETERS

---

<b>subaccountId</b>	integer <uint16>
	Filter for a subaccount.
<b>symbol</b>	string
	Market symbol to query. If not set, all markets are returned.
<b>limit</b>	integer <uint64>
	Maximum number to return. Default <a href="#">100</a> , maximum <a href="#">1000</a> .
<b>offset</b>	integer <uint64>
	Offset for pagination. Default <a href="#">0</a> .
<b>sortDirection</b>	string (SortDirection)
	Enum: <a href="#">"Asc"</a> <a href="#">"Desc"</a>

Sort direction.

---

## HEADER PARAMETERS

---

X-API-KEY	string API key
X-SIGNATURE	string Signature of the request
X-TIMESTAMP	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default <b>5000</b> , maximum <b>60000</b> )

---

## Responses

- **200** Success.
- **400** Bad request.
- **401** Unauthorized.
- **500** Internal server error.

GET /wapi/v1/history/funding

## Response samples

**200**    **400**    **401**    **500**

### Content type

application/json; charset=utf-8

[Copy](#)    [Expand all](#)    [Collapse all](#)

```
[  
  - {  
    "userId": 0,  
    "subaccountId": 0,  
    "symbol": "string",  
    "quantity": "string",  
    "intervalEndTimestamp": "string",  
    "fundingRate": "string"  
  }  
]
```

---

Get Liquidations History for ADL or Backstop Liquidations. This includes cases where the account was backstop or ADL liquidated, or when the account acted as a backstop or ADL provider for a liquidation.

#### QUERY PARAMETERS

<code>subaccountId</code>	integer <int32> Filter for a subaccount.
<code>type</code>	string (LiquidationSummaryEventType) Enum: <code>"BorrowLendADL"</code> <code>"BorrowLendADLProvider"</code> <code>"FuturesADL"</code> <code>"FuturesADLProvider"</code>
<code>symbol</code>	string Filter to the given symbol.
<code>limit</code>	integer <uint64> Maximum number to return. Default <code>100</code> , maximum <code>1000</code> .
<code>offset</code>	integer <uint64> Offset for pagination. Default <code>0</code> .
<code>sortDirection</code>	string (SortDirection) Enum: <code>"Asc"</code> <code>"Desc"</code> Sort direction.

#### Responses

> **200 Success.**

> **400 Bad request.**

> **401 Unauthorized.**

> **500 Internal server error.**

GET /wapi/v1/history/liquidations

## Response samples

**200**

**400**

**401**

**500**

### Content type

application/json; charset=utf-8

Copy    Expand all    Collapse all

```
[  
- {  
    "symbol": "string",  
    "netQuantityChange": "string",  
    "unsettledEquityChange": "string",  
    "markPrice": "string",  
    "liquidationPrice": "string",  
    "eventType": "BorrowLendADL",  
    "timestamp": "string"  
}  
]
```

## Get order history.

Retrieves the order history for the user. This includes orders that have been filled and are no longer on the book. It may include orders that are on the book, but the `/orders` endpoint contains more up to date data.

**Instruction:** `orderHistoryQueryAll`

---

QUERY PARAMETERS

orderId	string Filter to the given order.
strategyId	string Filter to the given strategy.
symbol	string Filter to the given symbol.
limit	integer <uint64> Maximum number to return. Default <b>100</b> , maximum <b>1000</b> .
offset	integer <uint64> Offset. Default <b>0</b> .
marketType	Array of strings (MarketType) Items Enum: <b>"SPOT"</b> <b>"PERP"</b> <b>"IPERP"</b> <b>"DATED"</b> <b>"PREDICTION"</b> <b>"RFQ"</b> Market type.
sortDirection	string (SortDirection) Enum: <b>"Asc"</b> <b>"Desc"</b> Sort direction.

---

## HEADER PARAMETERS

X-API-KEY	string API key
X-SIGNATURE	string Signature of the request
X-TIMESTAMP	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default <b>5000</b> , maximum <b>60000</b> )

---

## Responses

› **200 Success.**

> **400** Bad request.

> **401** Unauthorized.

> **500** Internal server error.

GET /wapi/v1/history/orders

### Response samples

200

400

401

500

#### Content type

application/json; charset=utf-8

Copy

Expand all

Collapse all

```
[  
  - {  
      "id": "string",  
      "createdAt": "string",  
      "executedQuantity": "string",  
      "executedQuoteQuantity": "string",  
      "expiryReason": "AccountTradingSuspended",  
      "orderType": "Market",  
      "postOnly": true,  
      "price": "string",  
      "quantity": "string",  
      "quoteQuantity": "string",  
      "selfTradePrevention": "RejectTaker",  
      "status": "Cancelled",  
      "side": "Bid",  
      "stopLossTriggerPrice": "string",  
      "stopLossLimitPrice": "string",  
      "stopLossTriggerBy": "string",  
      "symbol": "string",  
      "takeProfitTriggerPrice": "string",  
      "takeProfitLimitPrice": "string",  
      "takeProfitTriggerBy": "string",  
      "timeInForce": "GTC",  
      "triggerBy": "string",  
      "triggerPrice": "string",  
      "triggerQuantity": "string",  
      "clientId": 0,  
      "systemOrderType": "CollateralConversion",  
      "strategyId": "string",  
      "slippageTolerance": "string",  
      "slippageToleranceType": "TickSize"  
    }  
]
```

## Get rfq history.

Retrieves the rfq history for the user. This includes RFQs that have been filled or expired.

**Instruction:** [rfqHistoryQueryAll](#)

### QUERY PARAMETERS

<code>rfqId</code>	string	Filter to the given rfq.
<code>symbol</code>	string	Filter to the given symbol.
<code>status</code>	string (OrderStatus) Enum: "Cancelled" "Expired" "Filled" "New" "PartiallyFilled" "TriggerPending" "TriggerFailed"	Filter to the given status.
<code>side</code>	string (Side) Enum: "Bid" "Ask"	Filter to the given side.
<code>limit</code>	integer <uint64> Maximum number to return. Default <b>100</b> , maximum <b>1000</b> .	
<code>offset</code>	integer <uint64> Offset. Default <b>0</b> .	
<code>sortDirection</code>	string (SortDirection) Enum: "Asc" "Desc"	Sort direction.

## HEADER PARAMETERS

<code>X-API-KEY</code>	string API key
<code>X-SIGNATURE</code>	string Signature of the request
<code>X-TIMESTAMP</code>	integer <int64> Timestamp of the request in milliseconds
<code>X-WINDOW</code>	integer <uint64> Time the request is valid for in milliseconds (default <b>5000</b> , maximum <b>60000</b> )

## Responses

› **200 Success.**

› **400 Bad request.**

> **401** Unauthorized.

> **500** Internal server error.

GET /wapi/v1/history/rfq

## Response samples

200

400

401

500

### Content type

application/json; charset=utf-8

[Copy](#) [Expand all](#) [Collapse all](#)

```
[  
- {  
    "userId": 0,  
    "subaccountId": 0,  
    "rfqId": "string",  
    "clientId": 0,  
    "symbol": "string",  
    "side": "Bid",  
    "price": "string",  
    "quantity": "string",  
    "quoteQuantity": "string",  
    "submissionTime": "string",  
    "expiryTime": "string",  
    "status": "Cancelled",  
    "executionMode": "AwaitAccept",  
    "createdAt": "string"  
}  
]
```

## Get quote history.

Retrieves the quote history for the user. This includes quotes that have been filled or expired.

**Instruction:** `quoteHistoryQueryAll`

---

QUERY PARAMETERS

---

quoteId	string	
		Filter to the given quote.
symbol	string	
		Filter to the given symbol.
status	string (OrderStatus)	
	Enum:	"Cancelled" "Expired" "Filled" "New" "PartiallyFilled" "TriggerPending" "TriggerFailed"
		Filter to the given status.
limit	integer <uint64>	
		Maximum number to return. Default 100, maximum 1000.
offset	integer <uint64>	
		Offset. Default 0.
sortDirection	string (SortDirection)	
	Enum:	"Asc" "Desc"
		Sort direction.

---

HEADER PARAMETERS

---

X-API-KEY	string	
		API key
X-SIGNATURE	string	
		Signature of the request
X-TIMESTAMP	integer <int64>	
		Timestamp of the request in milliseconds
X-WINDOW	integer <uint64>	
		Time the request is valid for in milliseconds (default 5000, maximum 60000)

---

## Responses

> **200** Success.

> **400** Bad request.

> **401** Unauthorized.

> **500 Internal server error.**

GET /wapi/v1/history/quote

## Response samples

200

400

401

500

### Content type

application/json; charset=utf-8

[Copy](#) [Expand all](#) [Collapse all](#)

```
[  
- {  
    "userId": 0,  
    "subaccountId": 0,  
    "rfqId": "string",  
    "quoteId": "string",  
    "clientId": 0,  
    "bidPrice": "string",  
    "askPrice": "string",  
    "status": "Cancelled",  
    "createdAt": "string"  
}  
]
```

## Get settlement history.

History of settlement operations for the account.

**Instruction:** [settlementHistoryQueryAll](#)

### QUERY PARAMETERS

limit

integer <uint64>

Maximum number to return. Default [100](#), maximum [1000](#).

offset

integer <uint64>

Offset for pagination. Default `0`.

---

<code>source</code>	string (SettlementSourceFilter) Enum: <code>"BackstopLiquidation"</code> <code>"CulledBorrowInterest"</code> <code>"CulledRealizePnl"</code> <code>"CulledRealizePnlBookUtilization"</code> <code>"FundingPayment"</code> <code>"RealizePnl"</code> <code>"TradingFees"</code> <code>"TradingFeesSystem"</code>
---------------------	---

---

<code>sortDirection</code>	string (SortDirection) Enum: <code>"Asc"</code> <code>"Desc"</code> Sort direction.
----------------------------	---

---

## HEADER PARAMETERS

---

<code>X-API-KEY</code>	string API key
<code>X-SIGNATURE</code>	string Signature of the request
<code>X-TIMESTAMP</code>	integer <int64> Timestamp of the request in milliseconds
<code>X-WINDOW</code>	integer <uint64> Time the request is valid for in milliseconds (default <code>5000</code> , maximum <code>60000</code> )

---

## Responses

> **200 Success.**

> **400 Bad request.**

> **401 Unauthorized.**

> **500 Internal server error.**

GET /wapi/v1/history/settlement

## Response samples

[200](#)[400](#)[401](#)[500](#)**Content type**

application/json; charset=utf-8

[Copy](#)
[Expand all](#)
[Collapse all](#)

```
[
  - {
    "quantity": "string",
    "source": "TradingFees",
    "subaccountId": 0,
    "timestamp": "string",
    "userId": 0
  }
]
```

## Get strategy history.

Retrieves the strategy history for the user. This returns strategies that are no longer active as they have either been completed, cancelled by the user or cancelled by the system.

**Instruction:** [strategyHistoryQueryAll](#)

### QUERY PARAMETERS

**strategyId**

string

Filter to the given strategy.

**symbol**

string

Filter to the given symbol.

**limit**

integer &lt;uint64&gt;

Maximum number to return. Default [100](#), maximum [1000](#).**offset**

integer &lt;uint64&gt;

Offset. Default [0](#).**marketType**

Array of strings (MarketType)

Items Enum: ["SPOT"](#) ["PERP"](#) ["IPERP"](#) ["DATED"](#) ["PREDICTION"](#) ["RFQ"](#)

Market type.

**sortDirection**

string (SortDirection)

Enum:  

Sort direction.

**HEADER PARAMETERS**

**X-API-KEY** string  
API key

**X-SIGNATURE** string  
Signature of the request

**X-TIMESTAMP** integer <int64>  
Timestamp of the request in milliseconds

**X-WINDOW** integer <uint64>  
Time the request is valid for in milliseconds (default , maximum  )

**Responses**> **200** Success.> **400** Bad request.> **401** Unauthorized.> **500** Internal server error.

GET /wapi/v1/history/strategies

**Response samples****Content type**

application/json; charset=utf-8

```
[  
  - {  
      "id": "string",  
      "createdAt": "string",  
      "executedQuantity": "string",  
      "executedQuoteQuantity": "string",  
      "cancelReason": "Expired",  
      "strategyType": "Scheduled",  
      "quantity": "string",  
      "selfTradePrevention": "RejectTaker",  
      "status": "Running",  
      "side": "Bid",  
      "symbol": "string",  
      "timeInForce": "GTC",  
      "clientStrategyId": 0,  
      "duration": 0,  
      "interval": 0,  
      "randomizedIntervalQuantity": true,  
      "slippageTolerance": "string",  
      "slippageToleranceType": "TickSize"  
    }  
]
```

---

## Order

Order management.

### Get open order.

Retrieves an open order from the order book. This only returns the order if it is resting on the order book (i.e. has not been completely filled, expired, or cancelled).

One of `orderId` or `clientId` must be specified. If both are specified then the request will be rejected.

**Instruction:** `orderQuery`

---

#### QUERY PARAMETERS

<code>clientId</code>	integer <uint32>
	Client ID of the order.
<code>orderId</code>	string
	ID of the order.
<code>symbol</code> <b>required</b>	string
	Market symbol for the order.

## HEADER PARAMETERS

<code>X-API-KEY</code>	string
	API key
<code>X-SIGNATURE</code>	string
	Signature of the request
<code>X-TIMESTAMP</code>	integer <int64>
	Timestamp of the request in milliseconds
<code>X-WINDOW</code>	integer <uint64>
	Time the request is valid for in milliseconds (default <code>5000</code> , maximum <code>60000</code> )

## Responses

- **200** Success.
- **400** Bad request.
- **404** Order not found.
- **500** Internal server error

GET /api/v1/order

## Response samples

`200`

`400`

`404`

`500`

**Content type**

application/json; charset=utf-8

**Example**

Market

[Copy](#)

```
{  
    "orderType": "Market",  
    "id": "string",  
    "clientId": 0,  
    "createdAt": 0,  
    "executedQuantity": "string",  
    "executedQuoteQuantity": "string",  
    "quantity": "string",  
    "quoteQuantity": "string",  
    "reduceOnly": true,  
    "timeInForce": "GTC",  
    "selfTradePrevention": "RejectTaker",  
    "side": "Bid",  
    "status": "Cancelled",  
    "stopLossTriggerPrice": "string",  
    "stopLossLimitPrice": "string",  
    "stopLossTriggerBy": "string",  
    "symbol": "string",  
    "takeProfitTriggerPrice": "string",  
    "takeProfitLimitPrice": "string",  
    "takeProfitTriggerBy": "string",  
    "triggerBy": "string",  
    "triggerPrice": "string",  
    "triggerQuantity": "string",  
    "triggeredAt": 0,  
    "relatedOrderId": "string",  
    "strategyId": "string",  
    "slippageTolerance": "string",  
    "slippageToleranceType": "TickSize"  
}
```

---

## Execute order.

Submits an order to the matching engine for execution.

**Instruction:** `orderExecute`

## HEADER PARAMETERS

---

X-API-KEY <b>required</b>	string API key
X-SIGNATURE <b>required</b>	string Signature of the request
X-TIMESTAMP <b>required</b>	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default <code>5000</code> , maximum <code>60000</code> )
X-BROKER-ID	integer <uint16> Broker ID of the order
X-Broker-Id	integer <uint16>

REQUEST BODY SCHEMA: `application/json; charset=utf-8`

**required**

---

autoLend	boolean If true then the order can lend. Spot margin only.
autoLendRedeem	boolean If true then the order can redeem a lend if required. Spot margin only.
autoBorrow	boolean If true then the order can borrow. Spot margin only.
autoBorrowRepay	boolean If true then the order can repay a borrow. Spot margin only.
brokerId	integer <uint16> Broker ID of the order.
clientId	integer <uint32> Custom order id.
orderType <b>required</b>	string Enum: <code>"Market"</code> <code>"Limit"</code> Order type, market or limit.

<code>postOnly</code>	boolean	Only post liquidity, do not take liquidity.
<code>price</code>	string <decimal>	The order price if this is a limit order.
<code>quantity</code>	string <decimal>	The order quantity. Market orders must specify either a <code>quantity</code> or <code>quoteQuantity</code> . All other order types must specify a <code>quantity</code> .
<code>quoteQuantity</code>	string <decimal>	The maximum amount of the quote asset to spend (Ask) or receive (Bid) for market orders. This is used for reverse market orders. The order book will execute a <code>quantity</code> as close as possible to the notional value of <code>quoteQuantity</code> .
<code>reduceOnly</code>	boolean	If true then the order can only reduce the position. Futures only.
<code>selfTradePrevention</code>	string Enum: "RejectTaker"   "RejectMaker"   "RejectBoth"	Action to take if the user crosses themselves in the order book.
<code>side</code> <b>required</b>	string Enum: "Bid"   "Ask"	Order will be matched against the resting orders on the other side of the order book.
<code>stopLossLimitPrice</code>	string <decimal>	Stop loss limit price. If set the stop loss will be a limit order.
<code>stopLossTriggerBy</code>	string	Reference price that should trigger the stop loss order.
<code>stopLossTriggerPrice</code>	string	Stop loss price (price the stop loss order will be triggered at).
<code>symbol</code> <b>required</b>	string	The market for the order.
<code>takeProfitLimitPrice</code>	string <decimal>	Take profit limit price. If set the take profit will be a limit order,
<code>takeProfitTriggerBy</code>	string	Reference price that should trigger the take profit order.
<code>takeProfitTriggerPrice</code>	string	Take profit price (price the take profit order will be triggered at).

timeInForce	string
	Enum: "GTC" "IOC" "FOK"
	How long the order is good for.
triggerBy	string
	Trigger by.
triggerPrice	string
	Trigger price if this is a conditional order.
triggerQuantity	string
	Trigger quantity type if this is a trigger order.
slippageTolerance	string <decimal>
	Slippage tolerance allowed for the order.
slippageToleranceType	string
	Enum: "TickSize" "Percent"
	Slippage tolerance type.

## Responses

- > **200** Order executed.
- > **400** Bad request.
- > **500** Internal server error.
- > **503** System under maintenance.

POST /api/v1/order

## Request samples

Payload

Content type  
application/json; charset=utf-8

Copy

```
{
    "autoLend": true,
    "autoLendRedeem": true,
    "autoBorrow": true,
    "autoBorrowRepay": true,
    "brokerId": 0,
    "clientId": 0,
    "orderType": "Market",
    "postOnly": true,
    "price": "string",
    "quantity": "string",
    "quoteQuantity": "string",
    "reduceOnly": true,
    "selfTradePrevention": "RejectTaker",
    "side": "Bid",
    "stopLossLimitPrice": "string",
    "stopLossTriggerBy": "string",
    "stopLossTriggerPrice": "string",
    "symbol": "string",
    "takeProfitLimitPrice": "string",
    "takeProfitTriggerBy": "string",
    "takeProfitTriggerPrice": "string",
    "timeInForce": "GTC",
    "triggerBy": "string",
    "triggerPrice": "string",
    "triggerQuantity": "string",
    "slippageTolerance": "string",
    "slippageToleranceType": "TickSize"
}
```

## Response samples

[200](#)
[400](#)
[500](#)
[503](#)

### Content type

application/json; charset=utf-8

### Example

Market

[Copy](#)

```
{
    "orderType": "Market",
    "id": "string",
    "clientId": 0,
    "createdAt": 0,
```

```

    "executedQuantity": "string",
    "executedQuoteQuantity": "string",
    "quantity": "string",
    "quoteQuantity": "string",
    "reduceOnly": true,
    "timeInForce": "GTC",
    "selfTradePrevention": "RejectTaker",
    "side": "Bid",
    "status": "Cancelled",
    "stopLossTriggerPrice": "string",
    "stopLossLimitPrice": "string",
    "stopLossTriggerBy": "string",
    "symbol": "string",
    "takeProfitTriggerPrice": "string",
    "takeProfitLimitPrice": "string",
    "takeProfitTriggerBy": "string",
    "triggerBy": "string",
    "triggerPrice": "string",
    "triggerQuantity": "string",
    "triggeredAt": 0,
    "relatedOrderId": "string",
    "strategyId": "string",
    "slippageTolerance": "string",
    "slippageToleranceType": "TickSize"
}

```

## Cancel open order.

Cancels an open order from the order book.

One of `orderId` or `clientId` must be specified. If both are specified then the request will be rejected.

**Instruction:** `orderCancel`

### HEADER PARAMETERS

X-API-KEY <b>required</b>	string API key
------------------------------	-------------------

X-SIGNATURE <b>required</b>	string Signature of the request
--------------------------------	------------------------------------

X-TIMESTAMP  
**required**

integer <int64>  
Timestamp of the request in milliseconds

X-WINDOW

integer <uint64>  
Time the request is valid for in milliseconds (default **5000**, maximum **60000**)

REQUEST BODY SCHEMA: application/json; charset=utf-8  
**required**

clientId

integer <uint32>  
Client ID of the order.

orderId

string  
ID of the order.

symbol  
**required**

string  
Market the order exists on.

## Responses

> **200** Order cancelled.

— **202** Request accepted but not yet executed.

> **400** Bad request.

> **500** Internal server error.

> **503** System under maintenance.

DELETE /api/v1/order

## Request samples

**Payload**

**Content type**

application/json; charset=utf-8

[Copy](#)

```
{
  "clientId": 0,
  "orderId": "string",
  "symbol": "string"
}
```

## Response samples

[200](#)[400](#)[500](#)[503](#)

### Content type

application/json; charset=utf-8

### Example

Market

[Copy](#)

```
{
  "orderType": "Market",
  "id": "string",
  "clientId": 0,
  "createdAt": 0,
  "executedQuantity": "string",
  "executedQuoteQuantity": "string",
  "quantity": "string",
  "quoteQuantity": "string",
  "reduceOnly": true,
  "timeInForce": "GTC",
  "selfTradePrevention": "RejectTaker",
  "side": "Bid",
  "status": "Cancelled",
  "stopLossTriggerPrice": "string",
  "stopLossLimitPrice": "string",
  "stopLossTriggerBy": "string",
  "symbol": "string",
  "takeProfitTriggerPrice": "string",
  "takeProfitLimitPrice": "string",
  "takeProfitTriggerBy": "string",
  "triggerBy": "string",
  "triggerPrice": "string",
  "triggerQuantity": "string",
  "triggeredAt": 0,
  "relatedOrderId": "string",
  "strategyId": "string",
  "slippageTolerance": "string",
}
```

```

    "slippageToleranceType": "TickSize"
}

```

---

## Execute orders.

Submits a set of orders to the matching engine for execution in a batch.

**Batch commands instruction:** `orderExecute`

### HEADER PARAMETERS

X-API-KEY <b>required</b>	string API key
X-SIGNATURE <b>required</b>	string Signature of the request
X-TIMESTAMP <b>required</b>	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default <code>5000</code> , maximum <code>60000</code> )
X-Broker-Id	integer <uint16>

REQUEST BODY SCHEMA: `application/json; charset=utf-8`

**required**

Array [

autoLend	boolean If true then the order can lend. Spot margin only.
autoLendRedeem	boolean If true then the order can redeem a lend if required. Spot margin only.
autoBorrow	boolean If true then the order can borrow. Spot margin only.
autoBorrowRepay	boolean If true then the order can repay a borrow. Spot margin only.

<code>brokerId</code>	integer <uint16> Broker ID of the order.
<code>clientId</code>	integer <uint32> Custom order id.
<code>orderType</code> <small>required</small>	string Enum: <code>"Market"</code> <code>"Limit"</code> Order type, market or limit.
<code>postOnly</code>	boolean Only post liquidity, do not take liquidity.
<code>price</code>	string <decimal> The order price if this is a limit order.
<code>quantity</code>	string <decimal> The order quantity. Market orders must specify either a <code>quantity</code> or <code>quoteQuantity</code> . All other order types must specify a <code>quantity</code> .
<code>quoteQuantity</code>	string <decimal> The maximum amount of the quote asset to spend (Ask) or receive (Bid) for market orders. This is used for reverse market orders. The order book will execute a <code>quantity</code> as close as possible to the notional value of <code>quoteQuantity</code> .
<code>reduceOnly</code>	boolean If true then the order can only reduce the position. Futures only.
<code>selfTradePrevention</code>	string Enum: <code>"RejectTaker"</code> <code>"RejectMaker"</code> <code>"RejectBoth"</code> Action to take if the user crosses themselves in the order book.
<code>side</code> <small>required</small>	string Enum: <code>"Bid"</code> <code>"Ask"</code> Order will be matched against the resting orders on the other side of the order book.
<code>stopLossLimitPrice</code>	string <decimal> Stop loss limit price. If set the stop loss will be a limit order.
<code>stopLossTriggerBy</code>	string Reference price that should trigger the stop loss order.
<code>stopLossTriggerPrice</code>	string Stop loss price (price the stop loss order will be triggered at).
<code>symbol</code>	string

**required**

The market for the order.

**takeProfitLimitPrice**

string &lt;decimal&gt;

Take profit limit price. If set the take profit will be a limit order,

**takeProfitTriggerBy**

string

Reference price that should trigger the take profit order.

**takeProfitTriggerPrice**

string

Take profit price (price the take profit order will be triggered at).

**timeInForce**

string

Enum: "GTC" "IOC" "FOK"

How long the order is good for.

**triggerBy**

string

Trigger by.

**triggerPrice**

string

Trigger price if this is a conditional order.

**triggerQuantity**

string

Trigger quantity type if this is a trigger order.

**slippageTolerance**

string &lt;decimal&gt;

Slippage tolerance allowed for the order.

**slippageToleranceType**

string

Enum: "TickSize" "Percent"

Slippage tolerance type.

]

## Responses

> **200** Batch orders executed.

> **400** Bad request.

> **500** Internal server error.

> **503** System under maintenance.

POST /api/v1/orders

## Request samples

### Payload

#### Content type

application/json; charset=utf-8

[Copy](#) [Expand all](#) [Collapse all](#)

```
[  
- {  
    "autoLend": true,  
    "autoLendRedeem": true,  
    "autoBorrow": true,  
    "autoBorrowRepay": true,  
    "brokerId": 0,  
    "clientId": 0,  
    "orderType": "Market",  
    "postOnly": true,  
    "price": "string",  
    "quantity": "string",  
    "quoteQuantity": "string",  
    "reduceOnly": true,  
    "selfTradePrevention": "RejectTaker",  
    "side": "Bid",  
    "stopLossLimitPrice": "string",  
    "stopLossTriggerBy": "string",  
    "stopLossTriggerPrice": "string",  
    "symbol": "string",  
    "takeProfitLimitPrice": "string",  
    "takeProfitTriggerBy": "string",  
    "takeProfitTriggerPrice": "string",  
    "timeInForce": "GTC",  
    "triggerBy": "string",  
    "triggerPrice": "string",  
    "triggerQuantity": "string",  
    "slippageTolerance": "string",  
    "slippageToleranceType": "TickSize"  
}  
]
```

## Response samples

[200](#)[400](#)[500](#)[503](#)**Content type**

application/json; charset=utf-8

[Copy](#) [Expand all](#) [Collapse all](#)

```
[  
- {  
    "operation": "Ok",  
    "orderType": "Market",  
    "id": "string",  
    "clientId": 0,  
    "createdAt": 0,  
    "executedQuantity": "string",  
    "executedQuoteQuantity": "string",  
    "quantity": "string",  
    "quoteQuantity": "string",  
    "reduceOnly": true,  
    "timeInForce": "GTC",  
    "selfTradePrevention": "RejectTaker",  
    "side": "Bid",  
    "status": "Cancelled",  
    "stopLossTriggerPrice": "string",  
    "stopLossLimitPrice": "string",  
    "stopLossTriggerBy": "string",  
    "symbol": "string",  
    "takeProfitTriggerPrice": "string",  
    "takeProfitLimitPrice": "string",  
    "takeProfitTriggerBy": "string",  
    "triggerBy": "string",  
    "triggerPrice": "string",  
    "triggerQuantity": "string",  
    "triggeredAt": 0,  
    "relatedOrderId": "string",  
    "strategyId": "string",  
    "slippageTolerance": "string",  
    "slippageToleranceType": "TickSize"  
}  
]
```

---

**Get open orders.**

Retrieves all open orders. If a symbol is provided, only open orders for that market will be returned, otherwise all open orders are returned.

**Instruction:** [orderQueryAll](#)

## QUERY PARAMETERS

---

marketType	string (MarketType) Enum: <a href="#">"SPOT"</a> <a href="#">"PERP"</a> <a href="#">"IPERP"</a> <a href="#">"DATED"</a> <a href="#">"PREDICTION"</a> <a href="#">"RFQ"</a>
The market for the orders (SPOT or PERP).	

---

symbol	string The symbol of the market for the orders.
--------	--

---

## HEADER PARAMETERS

---

X-API-KEY	string API key
X-SIGNATURE	string Signature of the request
X-TIMESTAMP	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default <a href="#">5000</a> , maximum <a href="#">60000</a> )

---

## Responses

› [200 Success.](#)

› [400 Bad request.](#)

› [500 Internal Server Error.](#)

GET /api/v1/orders

## Response samples

[200](#)[400](#)[500](#)**Content type**

application/json; charset=utf-8

[Copy](#) [Expand all](#) [Collapse all](#)

```
[  
- {  
    "orderType": "Market",  
    "id": "string",  
    "clientId": 0,  
    "createdAt": 0,  
    "executedQuantity": "string",  
    "executedQuoteQuantity": "string",  
    "quantity": "string",  
    "quoteQuantity": "string",  
    "reduceOnly": true,  
    "timeInForce": "GTC",  
    "selfTradePrevention": "RejectTaker",  
    "side": "Bid",  
    "status": "Cancelled",  
    "stopLossTriggerPrice": "string",  
    "stopLossLimitPrice": "string",  
    "stopLossTriggerBy": "string",  
    "symbol": "string",  
    "takeProfitTriggerPrice": "string",  
    "takeProfitLimitPrice": "string",  
    "takeProfitTriggerBy": "string",  
    "triggerBy": "string",  
    "triggerPrice": "string",  
    "triggerQuantity": "string",  
    "triggeredAt": 0,  
    "relatedOrderId": "string",  
    "strategyId": "string",  
    "slippageTolerance": "string",  
    "slippageToleranceType": "TickSize"  
}  
]
```

## Cancel open orders.

Cancels all open orders on the specified market.

**Instruction:** `orderCancelAll`

## HEADER PARAMETERS

---

X-API-KEY <b>required</b>	string API key
X-SIGNATURE <b>required</b>	string Signature of the request
X-TIMESTAMP <b>required</b>	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default <code>5000</code> , maximum <code>60000</code> )

---

REQUEST BODY SCHEMA: `application/json; charset=utf-8`

**required**

symbol <b>required</b>	string Market to cancel orders for.
orderType	string Enum: <code>"RestingLimitOrder"</code> <code>"ConditionalOrder"</code> Type of orders to cancel.

---

## Responses

› **200** Success.

— **202** Request accepted but not yet executed.

› **400** Bad request.

› **500** Internal server error.

› **503** System under maintenance.

DELETE /api/v1/orders

## Request samples

### Payload

#### Content type

application/json; charset=utf-8

Copy

```
{  
  "symbol": "string",  
  "orderType": "RestingLimitOrder"  
}
```

## Response samples

200

400

500

503

#### Content type

application/json; charset=utf-8

Copy

Expand all

Collapse all

```
[  
- {  
    "orderType": "Market",  
    "id": "string",  
    "clientId": 0,  
    "createdAt": 0,  
    "executedQuantity": "string",  
    "executedQuoteQuantity": "string",  
    "quantity": "string",  
    "quoteQuantity": "string",  
    "reduceOnly": true,  
    "timeInForce": "GTC",  
    "selfTradePrevention": "RejectTaker",  
    "side": "Bid",  
    "status": "Cancelled",  
    "stopLossTriggerPrice": "string",  
    "stopLossLimitPrice": "string",  
    "stopLossTriggerBy": "string",  
    "symbol": "string",  
    "takeProfitTriggerPrice": "string",  
    "takeProfitLimitPrice": "string",  
    "takeProfitTriggerBy": "string",  
    "triggerBy": "string",  
    "triggerPrice": "string",  
    "triggerQuantity": "string",  
    "triggeredAt": 0,  
    "relatedOrderId": "string",  
    "strategyId": "string",  
    "slippageTolerance": "string",  
    "slippageToleranceType": "TickSize"  
}  
]
```

---

## Request For Quote

Request For Quote.

Submit RFQ.

Submit a Request for Quote (RFQ). The RFQ will be available for a specified time window for makers to respond to.

**Instruction:** `rfqSubmit`

## HEADER PARAMETERS

---

X-API-KEY <b>required</b>	string API key
X-SIGNATURE <b>required</b>	string Signature of the request
X-TIMESTAMP <b>required</b>	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default <code>5000</code> , maximum <code>60000</code> )

---

REQUEST BODY SCHEMA: `application/json; charset=utf-8`  
**required**

---

clientId	integer <uint32> Custom RFQ ID.
quantity	string <decimal> RFQ quantity (in base asset).
quoteQuantity	string <decimal> RFQ quote quantity (in quote asset).
price	string <decimal> RFQ price. Only when execution mode is <code>Immediate</code> .
symbol <b>required</b>	string RFQ symbol.
side <b>required</b>	string Enum: <code>"Bid"</code> <code>"Ask"</code> Side of the order.
executionMode	string Enum: <code>"AwaitAccept"</code> <code>"Immediate"</code> Execution mode. Defaults to <code>AwaitAccept</code> when not provided.

---

## Responses

> **200** Accepted.

> **400** Bad request.

> **500** Internal server error.

> **503** System under maintenance.

POST /api/v1/rfq

## Request samples

### Payload

#### Content type

application/json; charset=utf-8

Copy

```
{  
    "clientId": 0,  
    "quantity": "string",  
    "quoteQuantity": "string",  
    "price": "string",  
    "symbol": "string",  
    "side": "Bid",  
    "executionMode": "AwaitAccept"  
}
```

## Response samples

**200**

**400**

**500**

**503**

#### Content type

application/json; charset=utf-8

Copy

```
{  
    "rfqId": "string",  
    "clientId": 0,  
    "symbol": "string",
```

```

    "side": "Bid",
    "price": "string",
    "quantity": "string",
    "quoteQuantity": "string",
    "submissionTime": 0,
    "systemOrderType": "CollateralConversion",
    "expiryTime": 0,
    "status": "Cancelled",
    "executionMode": "AwaitAccept",
    "createdAt": 0
}

```

---

## Accept quote.

Accept a specific quote from a maker in response to an RFQ.

**Instruction:** `quoteAccept`

### HEADER PARAMETERS

X-API-KEY <b>required</b>	string API key
X-SIGNATURE <b>required</b>	string Signature of the request
X-TIMESTAMP <b>required</b>	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default <code>5000</code> , maximum <code>60000</code> )

REQUEST BODY SCHEMA: `application/json; charset=utf-8`  
**required**

rfqId	string RFQ ID.
clientId	integer <uint32> Custom RFQ ID.

quoteId	string
<b>required</b>	RFQ quote ID.

---

## Responses

- > **200** Accepted.
- > **400** Bad request.
- > **500** Internal server error.
- > **503** System under maintenance.

POST /api/v1/rfq/accept

## Request samples

### Payload

#### Content type

application/json; charset=utf-8

Copy

```
{  
  "rfqId": "string",  
  "clientId": 0,  
  "quoteId": "string"  
}
```

## Response samples

**200**

**400**

**500**

**503**

#### Content type

application/json; charset=utf-8

Copy

```
{  
  "rfqId": "string",
```

```

    "clientId": 0,
    "symbol": "string",
    "side": "Bid",
    "price": "string",
    "quantity": "string",
    "quoteQuantity": "string",
    "submissionTime": 0,
    "systemOrderType": "CollateralConversion",
    "expiryTime": 0,
    "status": "Cancelled",
    "executionMode": "AwaitAccept",
    "createdAt": 0
}

```

---

## Refresh RFQ.

Refresh a RFQ, extending the time window it is available for.

**Instruction:** [rfqRefresh](#)

### HEADER PARAMETERS

X-API-KEY <b>required</b>	string API key
X-SIGNATURE <b>required</b>	string Signature of the request
X-TIMESTAMP <b>required</b>	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default <a href="#">5000</a> , maximum <a href="#">60000</a> )

REQUEST BODY SCHEMA: application/json; charset=utf-8  
**required**

rfqId <b>required</b>	string RFQ ID. An RFQ can only be refreshed using the RFQ ID.
--------------------------	--

## Responses

> **200** Accepted.

> **400** Bad request.

> **500** Internal server error.

> **503** System under maintenance.

POST /api/v1/rfq/refresh

## Request samples

**Payload**

**Content type**

application/json; charset=utf-8

{

  "rfqId": "string"

}

Copy

## Response samples

**200**

**400**

**500**

**503**

**Content type**

application/json; charset=utf-8

{

  "rfqId": "string",  
  "clientId": 0,  
  "symbol": "string",  
  "side": "Bid",  
  "price": "string",  
  "quantity": "string",  
  "quoteQuantity": "string",  
  "submissionTime": 0,

Copy

```

    "systemOrderType": "CollateralConversion",
    "expiryTime": 0,
    "status": "Cancelled",
    "executionMode": "AwaitAccept",
    "createdAt": 0
}

```

---

## Cancel RFQ.

**Instruction:** `rfqCancel`

### HEADER PARAMETERS

X-API-KEY <b>required</b>	string API key
X-SIGNATURE <b>required</b>	string Signature of the request
X-TIMESTAMP <b>required</b>	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default <code>5000</code> , maximum <code>60000</code> )

REQUEST BODY SCHEMA: `application/json; charset=utf-8`  
**required**

<code>rfqId</code>	string RFQ ID.
<code>clientId</code>	integer <uint32> Custom RFQ ID.

## Responses

> **200 Accepted.**

> **400 Bad request.**

> **500** Internal server error.

> **503** System under maintenance.

POST /api/v1/rfq/cancel

## Request samples

### Payload

#### Content type

application/json; charset=utf-8

Copy

```
{  
  "rfqId": "string",  
  "clientId": 0  
}
```

## Response samples

200

400

500

503

#### Content type

application/json; charset=utf-8

Copy

```
{  
  "rfqId": "string",  
  "clientId": 0,  
  "symbol": "string",  
  "side": "Bid",  
  "price": "string",  
  "quantity": "string",  
  "quoteQuantity": "string",  
  "submissionTime": 0,  
  "systemOrderType": "CollateralConversion",  
  "expiryTime": 0,  
  "status": "Cancelled",  
  "executionMode": "AwaitAccept",  
  "createdAt": 0
```

}

## Submit quote.

Submit a quote in response to an RFQ. If valid, the quote may be accepted within the specified time window.

**Instruction:** `quoteSubmit`

### HEADER PARAMETERS

X-API-KEY <b>required</b>	string API key
X-SIGNATURE <b>required</b>	string Signature of the request
X-TIMESTAMP <b>required</b>	integer <int64> Timestamp of the request in milliseconds
X-WINDOW	integer <uint64> Time the request is valid for in milliseconds (default <code>5000</code> , maximum <code>60000</code> )

### REQUEST BODY SCHEMA:

**required**

rfqId <b>required</b>	string RFQ ID.
clientId	integer <uint32> Custom RFQ quote ID.
bidPrice <b>required</b>	string <decimal> Bid price.
askPrice <b>required</b>	string <decimal> Ask price.

## Responses

> **200 Accepted.**

> **400 Bad request.**

> **500 Internal server error.**

> **503 System under maintenance.**

POST /api/v1/rfq/quote

## Request samples

### Payload

#### Content type

application/json; charset=utf-8

Copy

```
{  
  "rfqId": "string",  
  "clientId": 0,  
  "bidPrice": "string",  
  "askPrice": "string"  
}
```

## Response samples

**200**

**400**

**500**

**503**

#### Content type

application/json; charset=utf-8

Copy

```
{  
  "rfqId": "string",  
  "quoteId": "string",  
  "clientId": 0,  
  "bidPrice": "string",  
  "askPrice": "string",  
  "status": "Cancelled",  
  "createdAt": 0
```

}

## Streams

### Usage

#### Subscribing

To use the websocket API, connect to `wss://ws.backpack.exchange`.

To subscribe to a stream with the name `stream` send a text frame over the websocket connection with the following JSON payload:

```
{  
  "method": "SUBSCRIBE",  
  "params": ["stream"]  
}
```

Similarly, to unsubscribe from a stream with the name `stream`:

```
{  
  "method": "UNSUBSCRIBE",  
  "params": ["stream"]  
}
```

You can subscribe or unsubscribe from multiple streams if you include more than one in the params field.

All data from streams is wrapped in a JSON object of the following form:

```
{  
  "stream": "<stream>",  
  ...  
}
```

```

    "data": "<payload>"
```

The following command can be used to test subscribing to a stream:

```

(sleep 1; \
echo '{"method": "SUBSCRIBE", "params": ["depth.SOL_USDC"]}' ;\
cat) | \
wscat -c wss://ws.backpack.exchange
```

The payloads for each stream type are outlined below.

## Timing

Timestamps are in microseconds (except for the K-line start and end times). The event timestamp is the time the event was emitted from the websocket server, and the engine timestamp is the time the event was generated by the matching engine.

If a message aggregates more than one event (for example, a depth message), the engine timestamp will be the timestamp of the last matching engine event.

## Keeping the connection alive

To keep the connection alive, a **Ping** frame will be sent from the server every 60s, and a **Pong** is expected to be received from the client. If a **Pong** is not received within 120s, a **Close** frame will be sent and the connection will be closed.

If the server is shutting down, a **Close** frame will be sent and then a grace period of 30s will be given before the connection is closed. The client should reconnect after receiving the **Close** frame. The client will be reconnected to a server that is not shutting down.

## Private

Subscribing to a private stream requires a valid signature generated from an ED25519 keypair. For stream subscriptions, the signature should be of the form:

```
instruction=subscribe&timestamp=1614550000000&window=5000
```

Where the timestamp and window are in milliseconds.

Private streams are prefixed with `account.` and require signature data to be submitted in the subscribe parameters. The verifying key and signature should be base64 encoded.

```
{
  "method": "SUBSCRIBE",
  "params": [ "stream" ],
  "signature": [ "<verifying key>", "<signature>", "<timestamp>", "<window>" ]
}
```

## Order update

On any mutation to an order the order will be pushed to the order update stream. The event type of the order update will be one of the following:

- `orderAccepted`
- `orderCancelled`
- `orderExpired`
- `orderFill`
- `orderModified`
- `triggerPlaced`
- `triggerFailed`

An `orderModified` update will be received when a resting reduce only order's quantity is decreased in order to prevent position side reversal.

## Stream Name Format

- For all markets: `account.orderUpdate`
- For single market: `account.orderUpdate.<symbol>`

```
{
  "e": "orderAccepted", // Event type
  "E": 1694687692980000, // Event time in microseconds
  "s": "SOL_USD", // Symbol
  "c": 123, // Client order ID
  "S": "Bid", // Side
  "o": "LIMIT", // Order type
  "f": "GTC", // Time in force
  "q": "32123", // Quantity
  "Q": "32123", // Quantity in quote
  "p": "20", // Price
  "P": "21", // Trigger price
  "B": "LastPrice", // Trigger by
}
```

```

    "a": "30",           // Take profit trigger price
    "b": "10",           // Stop loss trigger price
    "j": "30",           // Take profit limit price
    "k": "10",           // Stop loss limit price
    "d": "MarkPrice",   // Take profit trigger by
    "g": "IndexPrice",  // Stop loss trigger by
    "Y": "10",           // Trigger quantity
    "X": "Filled",       // Order state
    "R": "PRICE_BAND",  // Order expiry reason
    "i": "1111343026172067" // Order ID
    "t": 567,            // Trade ID
    "l": "1.23",          // Fill quantity
    "z": "321",           // Executed quantity
    "Z": "123",           // Executed quantity in quote
    "L": "20",             // Fill price
    "m": true,            // Whether the order was maker
    "n": "23",             // Fee
    "N": "USD",            // Fee symbol
    "V": "RejectTaker",   // Self trade prevention
    "T": 1694687692989999, // Engine timestamp in microseconds
    "O": "USER",           // Origin of the update
    "I": "1111343026156135" // Related order ID
    "H": 6023471188        // Strategy ID
    "Y": true              // Post only
}

```

There are several possible values for the `O` field (origin of the update):

- `USER`: The origin of the update was due to order entry by the user.
- `LIQUIDATION_AUTOCLOSE`: The origin of the update was due to a liquidation by the liquidation engine.
- `ADL_AUTOCLOSE`: The origin of the update was due to an ADL (auto-deleveraging) event.
- `COLLATERAL_CONVERSION`: The origin of the update was due to a collateral conversion to settle debt on the account.
- `SETTLEMENT_AUTOCLOSE`: The origin of the update was due to the settlement of a position on a dated market.
- `BACKSTOP_LIQUIDITY_PROVIDER`: The origin of the update was due to a backstop liquidity provider facilitating a liquidation.

Some fields are conditional on the order settings or event type:

- `c` - Only present if the order has a client order ID.
- `q` - Only present if the order has a quantity set.
- `Q` - Only present if the order is reverse market order.
- `p` - Only present if the order is a limit order.
- `P` - Only present if the order is a trigger order.
- `B` - Only present if the order is a trigger order.
- `a` - Only present if the order has a take profit trigger price set.
- `b` - Only present if the order has a stop loss trigger price set.
- `d` - Only present if the order has a take profit trigger price set.
- `g` - Only present if the order has a stop loss trigger price set.
- `Y` - Only present if the order is a trigger order.
- `R` - Only present if the event is a `orderExpired` event.
- `t` - Only present if the event is a `orderFill` event.
- `l` - Only present if the event is a `orderFill` event.
- `L` - Only present if the event is a `orderFill` event.

- `m` - Only present if the event is a `orderFill` event.
- `n` - Only present if the event is a `orderFill` event.
- `N` - Only present if the event is a `orderFill` event.

## Position update

On any mutation to a position the position will be pushed to the position update stream. The event type of the position update will be one of the following:

- `positionAdjusted`
- `positionOpened`
- `positionClosed`

On subscription, a message will be sent to the client with the current open positions, if any. The `e` field will not be present in the message.

## Stream Name Format

- For all markets: `account.positionUpdate`
- For single market: `account.positionUpdate.<symbol>`

```
{
  "e": "positionOpened", // Event type
  "E": 1694687692980000, // Event time in microseconds
  "s": "SOL_USDC_PERP", // Symbol
  "b": 123, // Break event price
  "B": 122, // Entry price
  "f": 0.5, // Initial margin fraction
  "M": 122, // Mark price
  "m": 0.01, // Maintenance margin fraction
  "q": 5, // Net quantity
  "Q": 6, // Net exposure quantity
  "n": 732, // Net exposure notional
  "i": "1111343026172067" // Position ID
  "p": "-1", // PnL realized
  "P": "0", // PnL unrealized
  "T": 1694687692989999 // Engine timestamp in microseconds
}
```

The net quantity field will be positive if the position is long and negative if the position is short.

The net exposure quantity field includes exposure from the open position, as well as any open orders.

## RFQ Update

This WebSocket stream provides real-time updates on RFQs (Request for Quotes) that are relevant to makers. Events are pushed to this stream whenever there is a significant state change in an RFQ or its associated quotes, allowing makers to monitor and respond to RFQs as they progress through various states.

## Event Types

For RFQs that submitted by other requesters.

- `rfqActive`: Indicates that an RFQ is active and open for quotes.

For RFQs that submitted by your account.

- `rfqAccepted`: Indicates that an RFQ has been accepted and is no longer active.
- `rfqRefreshed`: Indicates that an RFQ has been refreshed, is active and open for quotes.
- `rfqCancelled`: Indicates that an RFQ has been cancelled or expired.
- `rfqCandidate`: RFQ has received a new best quote.
- `rfqFilled`: Indicates that an RFQ has been fully filled with a quote.

For Quotes submitted by your account.

- `quoteAccepted`: Indicates that a quote submitted by the maker has been accepted.
- `quoteCancelled`: Indicates that a quote has been cancelled due to quote submission, RFQ being filled, refreshed, cancelled, or expired.

## Quote Submission and RFQ Timing

Makers should submit quotes before the **submission time** (`w` field) is reached, as indicated in each `rfqActive` event. An RFQ remains active until the **expiration time** (`w` field). If no quote is accepted or the RFQ is not cancelled, makers may continue to submit quotes until expiration.

RFQs can periodically request new quotes by issuing additional `rfqActive` events. Each new `rfqActive` event will have the same RFQ ID (`R` field) but updated values for **submission time** and **expiration time**, allowing makers to participate in extended or renewed quoting periods for ongoing RFQs.

## Stream Name Format

- For all markets: `account.rfqUpdate`
- For single market: `account.rfqUpdate.<symbol>`

## Example Messages

### RFQ Accepted (sent to requester)

```
{
  "e": "rfqAccepted",           // Event type
  "E": 1730225420369829,       // Event time in microseconds
  "R": 113392053149171712,     // RFQ ID
  "C": "123",                  // Client RFQ ID
  "s": "SOL_USDC_RFQ",         // Symbol
  "S": "Bid",                  // RFQ side
  "q": "10",                   // Quantity (if quantity in base asset)
  "w": 1730225480368,          // Submission time in milliseconds
  "W": 1730225540368,          // Expiry time in milliseconds
}
```

```

    "X": "New",           // RFQ status
    "T": 1730225420368765 // Engine timestamp in microseconds
}

```

**RFQ Active** (broadcast to all rfq listeners)

```

{
    "e": "rfqActive",      // Event type
    "E": 1730225420369829, // Event time in microseconds
    "R": 113392053149171712, // RFQ ID
    "s": "SOL_USDC_RFQ", // Symbol
    "q": "10",             // Quantity (optional) (if quantity in base as
    "w": 1730225480368,   // Submission time in milliseconds
    "W": 1730225540368,   // Expiry time in milliseconds
    "X": "New",            // RFQ status
    "T": 1730225420368765 // Engine timestamp in microseconds
}

```

**RFQ Refreshed** (sent to requester)

```

{
    "e": "rfqRefreshed", // Event type
    "E": 1730225450369829, // Event time in microseconds
    "R": 113392053149171712, // RFQ ID
    "C": "123",             // Client RFQ ID
    "s": "SOL_USDC_RFQ", // Symbol
    "S": "Bid",             // RFQ side
    "q": "10",              // Quantity (optional) (if quantity in base as
    "w": 1730225480368,   // Submission time in milliseconds
    "W": 1730225540368,   // Expiry time in milliseconds
    "X": "New",            // RFQ status
    "T": 1730225450368765 // Engine timestamp in microseconds
}

```

**RFQ Cancelled** (sent to taker only)

```

{
    "e": "rfqCancelled", // Event type
    "E": 1730225460369829, // Event time in microseconds
    "R": 113392053149171712, // RFQ ID
    "C": "123",             // Client RFQ ID
    "s": "SOL_USDC_RFQ", // Symbol
    "S": "Bid",             // RFQ side
    "Q": "150",              // Quote quantity (optional) (if quantity in q
    "w": 1730225480368,   // Submission time in milliseconds
    "W": 1730225540368,   // Expiry time in milliseconds
    "X": "Cancelled", // RFQ status
    "T": 1730225460368765 // Engine timestamp in microseconds
}

```

**Quote Accepted** (sent to quoter)

```
{
  "e": "quoteAccepted",           // Event type
  "E": 1730225434631394,        // Event time in microseconds
  "R": 113392053149171712,      // RFQ ID
  "u": 113392054083780608,      // Quote ID
  "C": "123",                   // Client Quote ID
  "s": "SOL_USDC_RFQ",          // Symbol
  "X": "New",                   // Quote status
  "T": 1730225434629778        // Engine timestamp in microseconds
}
```

**Quote Cancelled** (sent to quoter)

```
{
  "e": "quoteCancelled",         // Event type
  "E": 1730225583761963,        // Event time in microseconds
  "R": 113392061354344448,      // RFQ ID
  "u": 113392062870847488,      // Quote ID
  "C": "123",                   // Client Quote ID
  "s": "SOL_USDC_RFQ",          // Symbol
  "X": "Cancelled",             // Quote status
  "T": 1730225583753811        // Engine timestamp in microseconds
}
```

**RFQ Candidate** (sent to requester with quote details)

```
{
  "e": "rfqCandidate",           // Event type
  "E": 1730225490648996,        // Event time in microseconds
  "R": 113392053149171712,      // RFQ ID
  "u": 113392054083780608,      // Quote ID
  "C": "123",                   // Client RFQ ID
  "s": "SOL_USDC_RFQ",          // Symbol
  "S": "Bid",                   // RFQ side
  "q": "10",                    // RFQ quantity (in base asset)
  "Q": "150",                   // RFQ quote quantity (in quote asset)
  "p": "15.50",                 // Taker price (quote price + fee)
  "X": "New",                   // RFQ status
  "T": 1730225490647080        // Engine timestamp in microseconds
}
```

**RFQ Filled** (sent to both requester and quoter)

```
// To requester
{
  "e": "rfqFilled",              // Event type
  "E": 1730225497648996,        // Event time in microseconds
  "R": 113392053149171712,      // RFQ ID
  "u": 113392054083780608,      // Quote ID
  "C": "123",                   // Client RFQ ID
  "s": "SOL_USDC_RFQ",          // Symbol
  "S": "Bid",                   // RFQ side
  "Q": "150",                   // RFQ quote quantity (optional) (if quantity
  "p": "15.50",                 // Taker price (quote price + fee)
```

```

    "X": "Filled",           // RFQ status
    "T": 1730225497647080   // Engine timestamp in microseconds
}

// To quoter
{
    "e": "rfqFilled",      // Event type
    "E": 1730225497648996, // Event time in microseconds
    "R": 113392053149171712, // RFQ ID
    "u": 113392054083780608, // Quote ID
    "C": "123",             // Client Quote ID
    "s": "SOL_USDC_RFQ",   // Symbol
    "p": "15.00",            // Price
    "X": "Filled",          // Quote status
    "T": 1730225497647080   // Engine timestamp in microseconds
}

```

## Field Descriptions

- **e** - Event type (e.g., `rfqActive`, `rfqAccepted`, `rfqRefreshed`, `rfqCancelled`, `quoteAccepted`, `quoteCancelled`, `rfqCandidate`, `rfqFilled`).
- **E** - Event time in microseconds.
- **R** - RFQ ID, identifying the request for quote.
- **u** - Quote ID, identifying the specific quote.
- **C** - Client ID (either Client RFQ ID or Client Quote ID depending on context).
- **s** - Symbol the RFQ is for.
- **S** - Side of the RFQ, either "Bid" or "Ask".
- **q** - Quantity for the RFQ (in base asset, if quantity in base asset).
- **Q** - Quote quantity for the RFQ (in quote asset, if quantity in quote asset).
- **p** - Price associated with the quote/fill event.
- **w** - Submission time for the RFQ in milliseconds.
- **W** - Expiry time for the RFQ in milliseconds.
- **X** - Order status (e.g., `New`, `Cancelled`, `Filled`).
- **T** - Engine timestamp in microseconds.

Some fields are conditional and may be present only in specific events.

## Public

### Book ticker

Stream name format: `bookTicker.<symbol>`

```
{
  "e": "bookTicker",           // Event type
  "E": 1694687965941000,     // Event time in microseconds
  "s": "SOL_USDC",           // Symbol
  "a": "18.70",               // Inside ask price
  "A": "1.000",               // Inside ask quantity
  "b": "18.67",               // Inside bid price
  "B": "2.000",               // Inside bid quantity
  "u": "111063070525358080", // Update ID of event
  "T": 1694687965940999     // Engine timestamp in microseconds
}
```

## Depth

Contains incremental depth updates. Each depth update has the absolute value of the depths at the given levels, and only changes when the depth has changed.

To obtain an initial snapshot of the depth, the client should query the [REST API](#).

The depth stream will push updates as quickly as possible, but under load it may aggregate more than one update into a single event. In this case the `U` and `u` fields will not be the same. The `U` field is the first update ID in the event, and the `u` field is the final update ID in the event.

There are alternative depth streams that aggregates updates into a single message over a 200ms, 600ms or 1000ms period instead of pushing updates in realtime. This is useful for reducing network traffic.

Updates are sequential, so `U` will always be `u + 1` from the previous message. If this is not the case, the client should assume that the depth has been invalidated and requery the REST API.

Stream name format: `depth.<symbol>` (realtime) Stream name format: `depth.200ms.<symbol>`  
 (aggregated) Stream name format: `depth.600ms.<symbol>` (aggregated) Stream name format:  
`depth.1000ms.<symbol>` (aggregated)

```
{
  "e": "depth",                // Event type
  "E": 1694687965941000,      // Event time in microseconds
  "s": "SOL_USDC",            // Symbol
  "a": [                        // Asks
    [
      "18.70",                 // Price
      "0.000"                  // Quantity
    ]
  ],
  "b": [                        // Bids
    [
      "18.67",                 // Price
      "0.832"                  // Quantity
    ],
    [
      "18.68",                 // Price
      "0.000"                  // Quantity
    ]
  ]
}
```

```

        "0.000"
    ]
},
"U": 94978271,           // First update ID in event
"u": 94978271,           // Last update ID in event
"T": 1694687965940999  // Engine timestamp in microseconds
}

```

## K-Line

Stream name format: `kline.<interval>.<symbol>`

```

{
  "e": "kline",           // Event type
  "E": 1694687692980000, // Event time in microseconds
  "s": "SOL_USD",         // Symbol
  "t": "2024-09-11T12:00:00", // K-Line start time (ISO 8601 format)
  "T": "2024-09-11T12:01:00", // K-Line close time (ISO 8601 format)
  "o": "18.75",           // Open price
  "c": "19.25",           // Close price
  "h": "19.80",           // High price
  "l": "18.50",           // Low price
  "v": "32123",           // Base asset volume
  "n": 93828,             // Number of trades
  "X": false              // Is this k-line closed?
}

```

## Liquidation

Contains updates for liquidation events for all liquidation types.

Stream name format: `liquidation`

```

{
  "e": "liquidation",      // Event type
  "E": 1694688638091000, // Event time in microseconds
  "q": "10",               // Quantity
  "p": "18.70",             // Price
  "S": "Bid",               // Side
  "s": "SOL_USDC",          // Symbol
  "T": 567,                 // Engine timestamp in microseconds
}

```

## Mark price

Stream name format: `markPrice.<symbol>`

```
{
  "e": "markPrice",           // Event type
  "E": 1694687965941000,    // Event time in microseconds
  "s": "SOL_USDC",          // Symbol
  "p": "18.70",              // Mark price
  "f": "1.70",                // Estimated funding rate
  "i": "19.70",              // Index price
  "n": 1694687965941,       // Next funding timestamp in milliseconds
  "T": 1694687965940999    // Engine timestamp in microseconds
}
```

## Ticker

The ticker stream pushes 24hr rolling statistics for a single symbol every second.

Stream name format: `ticker.<symbol>`

```
{
  "e": "ticker",             // Event type
  "E": 1694687692980000,    // Event time in microseconds
  "s": "SOL_USD",            // Symbol
  "o": "18.75",               // First price
  "c": "19.24",               // Last price
  "h": "19.80",               // High price
  "l": "18.50",               // Low price
  "v": "32123",                // Base asset volume
  "V": "928190",                // Quote asset volume
  "n": 93828                  // Number of trades
}
```

## Open interest

Open interest updates are pushed to the `openInterest` stream every 60 seconds.

Stream name format: `openInterest.<symbol>`

```
{  
  "e": "openInterest",           // Event type  
  "E": 1694687965941000,        // Event time in microseconds  
  "s": "SOL_USDC_PERP",         // Symbol  
  "o": "100",                   // Open interest in contracts  
}
```

## Trade

Contains public trade data for a single symbol. The trade ID is a sequential number specific to the symbol. This stream includes updates for trades executed as a result of liquidations.

Stream name format: `trade.<symbol>`

```
{  
  "e": "trade",                 // Event type  
  "E": 1694688638091000,        // Event time in microseconds  
  "s": "SOL_USDC",              // Symbol  
  "p": "18.68",                 // Price  
  "q": "0.122",                 // Quantity  
  "b": "111063114377265150",    // Buyer order ID  
  "a": "111063114585735170",    // Seller order ID  
  "t": 12345,                   // Trade ID  
  "T": 1694688638089000,        // Engine timestamp in microseconds  
  "m": true                     // Is the buyer the maker?  
}
```

