

# WebSocket Protocol

[Jump to bottom](#)

haipo yang edited this page on Nov 27, 2017 · 53 revisions

## API protocol

The API is based on [JSON RPC](#) of Websocket protocol. Repeated subscription will be cancelled for the same data type.

### Request

- method: method , String
- params: parameters , Array
- id: Request id, Integer

### Response

- result: Json object , null for failure
- error: Json object , null for success, non-null for failure
  1. code: error code
  2. message: error message
- id: Request id, Integer

### Notify

- method: method , String
- params: parameters , Array
- id: Null

### General error code:

- 1: invalid argument
- 2: internal error
- 3: service unavailable
- 4: method not found
- 5: service timeout
- 6: require authentication

# System API

---

## PING

- method: `server.ping`
- params: None
- result: "pong"
- example: `{"error": null, "result": "pong", "id": 1000}`

## System time

- method: `server.time`
- params: none
- result: timestamp , Integer
- example: `{"error": null, "result": 1493285895, "id": 1000}`

## ID verification Web

- method: `server.auth`
- params:
  1. token: String
  2. source: String, source , e.g. "web", version number up to 30 bytes is required for applications

## ID verification Api

- method: `server.sign`
- params:
  1. access\_id: String
  2. authorization: String, sign data
  3. tonce: timestamp , for milliseconds spent from Unix epoch to current time , and error between tonce and server time can not exceed plus or minus 60s

# Market API

---

## Market inquiry

- method: `kline.query`
- params:
  1. market: market name
  2. start: start time , Integer
  3. end: end time, Integer
  4. interval: interval, Integer
- result:

```
"result": [
  [
    1492358400, time
    "7000.00", open
    "8000.0", close
    "8100.00", highest
    "6800.00", lowest
    "1000.00" volume
    "123456.00" amount
    "BTCCNY" market name
  ]
  ...
]
```

## Market subscription

- method: `kline.subscribe`
- params:

1. market
2. interval

## Market notification

- method: `kline.update`
- params:

```
[
  [
    1492358400, time
    "7000.00", open
    "8000.0", close
    "8100.00", highest
    "6800.00", lowest
    "1000.00" volume
    "123456.00" amount
    "BTCCNY" market name
  ]
  ...
]
```

## Cancel subscription

- method: `kline.unsubscribe`
- params: none

# Price API

Acquire latest price

- method: `price.query`
- params:

1. market: String, market name

- result: String, price

### Latest price subscription

- method: `price.subscribe`
- params: market list

### Latest market notification

- method: `price.update`
- params:

1. market: String

2. price: String

### Cancel subscription

- method: `price.unsubscribe`
- params: none

## Market status API

---

### Acquire market status

- method: `state.query`
  - params:
1. market: market
2. period: cycle period , Integer, e.g. 86400 for last 24 hours
- result:

```
{  
  "open": open price  
  "last": latest price  
  "high": highest price  
  "low": lowest price  
  "deal": amount  
  "volume": volume  
}
```

### Market 24H status subscription

- method: `state.subscribe`
- params: market list

## Market 24H status notification

- method: `state.update`
- params:

1. market:

2. result: same as query

## Cancel subscription

- method: `state.unsubscribe`
- params: none

# Market status in Today API

---

## Acquire Market today status

- method: `today.query`
- params:

1. market: market

- result:

```
{
  "open": Today open price
  "last": Today latest price
  "high": Today highest price
  "low":  Today lowest price
  "deal": 24H amount
  "volume": 24H volume
}
```

## Market today status subscription

- method: `today.subscribe`
- params: market list

## Market Today status notification

- method: `today.update`
- params:

1. market:

2. result: same as query

## Cancel subscription

- method: `today.unsubscribe`
- params: none

# Deal API

---

## Acquire latest executed list

- method: `deals.query`
- params:
  1. market : market
  2. limit : amount limit
  3. last\_id: largest ID of last returned result

## Latest order list subscription

- method: `deals.subscribe`
- params: market list

## Latest order list update

- method: `deals.update`
- params:
  1. market name
  2. order list

## Cancel subscription

- method: `deals.unsubscribe`
- params: none

# Depth API

---

## Acquire depth

- method: `depth.query`
- params:
  1. market : market name
  2. limit: amount limit , Integer
  3. interval: interval , String, e.g. "1" for 1 unit interval, "0" for no interval

```
"result": {  
  "asks": [  
    [  
      "8000.00",  
      "9.6250"  
    ]  
  ],  
  "bids": [  
    [  
      "7999.99",  
      "9.6250"  
    ]  
  ]  
}
```

```
        "7000.00",  
        "0.1000"  
    ]  
]  
}
```

## Depth subscription

- method: `depth.subscribe`
- params:
  1. market
  2. limit
  3. interval

## Depth notification

- method: `depth.update`
- params:
  1. clean: Boolean, true: complete result , false: last returned updated result
  2. Same as `depth.query` , only return what's different from last result, asks 或 bids may not exist. amount == 0 for delete
  3. market name

## Cancel subscription

- method: `depth.unsubscribe`
- params: none

# Order API (Authentication required before connection)

---

## Unexecuted order inquiry

- method: `order.query`
- params:
  1. market: market name , String
  2. offset: offset , Integer
  3. limit: limit , Integer
- result: see HTTP protocol

## Executed order inquiry

- method: `order.history`
- params:
  1. market: market name , String

2. start\_time: start time , 0 for unlimited , Integer
3. end\_time: end time , 0 for unlimited, Integer
4. offset: offset , Integer
5. limit: limit , Integer
6. side: side , 0 for unlimited , 1 for sell , 2 for buy

## Order subscription

- method: `order.subscribe`
- params: market list

## Order notification

- method: `order.update`
- params:
  1. event: event type , Integer, 1: PUT, 2: UPDATE, 3: FINISH
  2. order: order detail , Object , see HTTP protocol

## Cancel subscription

- method: `order.unsubscribe`
- params: none

# Asset API (Authentication required before connection)

---

## Asset inquiry

- method: `asset.query`
- params: asset list, null for inquire all
- result:

```
{"BTC": {"available": "1.10000000","freeze": "9.90000000"}}
```

## Asset history

- method: `asset.history`
- params:
  1. asset: asset name, which can be null
  2. business: business , which can be null, use ',' to separate types
  3. start\_time: start time, 0 for unlimited , Integer
  4. end\_time: end time, 0 for unlimited, Integer
  5. offset: offset , Integer
  6. limit: amount limit , Integer

## Asset subscription



- method: `asset.subscribe`
- params: asset list

Asset notification

- method: `asset.update`
- params:

[{"BTC": {"available": "1.10000000","freeze": "9.90000000"}, "CNY": {}}]

Cancel subscription

- method: `asset.unsubscribe`
- params: none

▼ Pages 5

Find a page...

▶ Home

▶ Architecture

▶ HTTP Protocol

▶ requirements

▼ WebSocket Protocol

- API protocol
- System API
- Market API
- Price API
- Market status API
- Market status in Today API
- Deal API
- Depth API
- Order API (Authentication required before connection)
- Asset API (Authentication required before connection)

Clone this wiki locally

https://github.com/viabtc/viabtc\_exchange\_server.wiki.git