**WebSocket API**

WebSocket Protocol 是HTML5一种新的协议(Protocol)。它是实现了客户端与服务器全双工通信，使得数据可以快速的双向传播。通过一次简单的握手就可以建立客户端和服务器连接，服务器根据业务规则可以主动推送信息给客户端。其优点如下：

* 客户端和服务器端之间数据传输时请求头信息比较小,大概2个字节。
* 服务器和客户端可以相互主动的发送数据给对方。
* 不需要多次创建TCP请求和销毁，节约宽带和服务器的资源。

如果在使用过程中有任何问题，请联系我们技术QQ群： 334662836，我们将为您做出最权威的解答。

**Getting Started**

* [安全认证](https://www.okcoin.com/about/ws_authentication.do)
* [请求交互说明](https://www.okcoin.com/about/ws_request.do)
* [现货行情 API](https://www.okcoin.com/about/ws_api.do#spapi)
* [现货交易 API](https://www.okcoin.com/about/ws_api.do#stapi)
* [期货行情 API](https://www.okcoin.com/about/ws_api.do#fpapi)
* [期货交易 API](https://www.okcoin.com/about/ws_api.do#ftapi)
* **获取API认证的apiKey和secretKey**
* API认证具体申请位置：基本设置->我的API，点击申请API即可获得，其中apiKey是OKCoin提供给API用户的访问密钥，secretKey用于对请求参数签名的私钥。 注意： 请勿向任何人泄露这两个参数，这两个参数关乎账号安全。
* **生成待签名字符串**
* 用户提交的参数除sign外，都要参与签名。待签名字符串要求按照参数名进行排序（首先比较所有参数名的第一个字母，按abcd顺序排列，若遇到相同首字母,则看第二个字母, 以此类推。）
* 例如：对于如下的参数进行签名 websocket.send("{'event':'addChannel','channel':'channel','parameters':{'api\_key':'c821db84-6fbd-11e4-a9e3-c86000d26d7c','sign':'4CBB1D1518F8BEE4040CE6B14F225C82','symbol':'btc\_usd','type':'buy','price':'50','amount':'0.02'}}")  
  生成待签名字符串为：amount=1.0&api\_key=c821db84-6fbd-11e4-a9e3-c86000d26d7c&price=680&symbol=btc\_usd&type=buy
* **MD5签名**
* 在MD5签名时,需要私钥secretKey参与签名。将待签名字符串添加私钥参数生成最终待签名字符串，例如：amount=1.0&api\_key=c821db84-6fbd-11e4-a9e3-c86000d26d7c&price=680&symbol=btc\_usd&type=buy&secret\_key=secretKey 注意“&secret\_key=secretKey” 为签名必传参数。利用32位MD5算法 对最终待签名字符串进行签名运算,从而得到签名结果字符串(该字符串赋值于参数 sign)，MD5计算结果中字母全部大写。
* **WebSocket服务地址**
* OKCoin WebSocket服务连接地址：wss://real.okcoin.com:10440/websocket/okcoinapi
* **发送请求**
* 请求数据格式为：{'event':'addChannel','channel':'channelValue','parameters':{'api\_key':'value1','sign':'value2'}}   
  其中event: addChannel(注册请求数据)/removeChannel(注销请求数据)   
  channel：OKCoin 提供请求数据类型   
  parameters 参数为选填参数，   
  其中api\_key 为用户申请apiKey,sign 为签名字符串，签名规则参照请求说明  
  例如：websocket.send("{'event':'addChannel','channel':'ok\_btcusd\_ticker'}")  
  websocket.send("[{'event':'addChannel','channel':'ok\_btcusd\_ticker'},{'event':'addChannel','channel':'ok\_btcusd\_depth'},{'event':'addChannel','channel':'ok\_btcusd\_trades'}]");支持批量注册。
* **服务器响应**
* 返回数据格式为： [{"channel":"channel","success":"","errorcode":"","data":{}}, {"channel":"channel","success":"","errorcode":1,"data":{}}]   
  channel:请求的数据类型  
  result:true成功,false失败(用于WebSocket 交易API)  
  data：返回结果数据  
  errorcode:错误码(用于WebSocket 交易API)

|  |  |
| --- | --- |
| **错误代码** | **详细描述** |
| 10001 | 参数错误 |
| 10002 | 验证失败 |
| 10003 | 该连接已经请求了其他用户的实时交易数据 |
| 10004 | 该连接没有请求此用户的实时交易数据 |
| 10005 | 系统错误 |
| 10009 | 订单不存在 |
| 10010 | 余额不足 |
| 10011 | 卖的数量小于BTC/LTC最小买卖额度 |
| 10012 | 当前网站暂时只支持btc\_usd ltc\_usd |
| 10014 | 下单价格不得≤0或≥1000000 |
| 10015 | 暂不支持此channel订阅 |
| 10016 | 币数量不足 |
| 10017 | WebSocket鉴权失败 |
| 10100 | 用户被冻结 |
| 10216 | 非开放API |
| 20001 | 用户不存在 |
| 20002 | 用户被冻结 |
| 20003 | 用户被爆仓冻结 |
| 20004 | 期货账户被冻结 |
| 20005 | 用户期货账户不存在 |
| 20006 | 必填参数为空 |
| 20007 | 参数错误 |
| 20008 | 期货账户余额为空 |
| 20009 | 期货合约状态错误 |
| 20010 | 期货风险率信息不存在 |
| 20011 | 开仓前保证金率超过90% |
| 20012 | 开仓后保证金率超过90% |
| 20013 | 暂无对手价 |
| 20014 | 系统错误 |
| 20015 | 订单信息不存在 |
| 20016 | 平仓数量是否大于同方向可用持仓数量 |
| 20017 | 非本人操作 |
| 20018 | 下单价格高于前一分钟的105%或低于95% |
| 20019 | 该IP限制不能请求该资源 |
| 20020 | 密钥不存在 |
| 20021 | 指数信息不存在 |
| 20022 | 接口调用错误 |
| 20023 | 逐仓用户 |
| 20024 | sign签名不匹配 |
| 20025 | 杠杆比率错误 |

* **推送过程说明**
* 为保证推送的及时性及减少流量，行情数据（ticker）和委托深度（depth）这两种数据类型只会在数据发生变化的情况下才会推送数据，交易记录（trades）是推送从上次推送到本次推送产生的增量数据。
* **如何判断连接是否断开？**
* OKCoin通过心跳机制解决这个问题。客户端发送的心跳数据：{'event':'ping'}，服务器会响应客户端：{"event":"pong"}以此来表明客户端和服务端保持正常连接。如果客户端未接到服务端响应的心跳数据则需要客户端重新建立连接。
* **现货行情 API**
* **获取OKCoin最新市场现货行情数据**

|  |  |
| --- | --- |
| **接口** | **描述** |
| ok\_btcusd\_ticker比特币行情数据  websocket.send("{'event':'addChannel','channel':'ok\_btcusd\_ticker'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_btcusd\_ticker'}  # Response  [{  "channel":" ok\_btcusd\_ticker",  "data":  {"buy":"2478.3",  "high":"2555",  "last":"2478.51",  "low":"2466",  "sell":"2478.5",  "timestamp":"1411718074965",  "vol":"49,020.30"}  }]    **返回值说明**  buy: 买一价  high: 最高价  last: 最新成交价  low: 最低价  sell: 卖一价  timestamp：时间戳  vol: 成交量(最近的24小时) | |
| ok\_btcusd\_depth比特币20条市场深度  websocket.send("{'event':'addChannel','channel':'ok\_btcusd\_depth'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_btcusd\_depth'}  # Response  [{  "channel":" ok\_btcusd\_depth",  "data":{  "bids":[[2473.88,2.025],  [2473.5,2.4],  [2470,12.203]],  "asks":[[2484,17.234],  [2483.01,6],  [2482.88,3]],  "timestamp":"1411718972024"  }  }]    **返回值说明**  bids:买方深度  asks:卖方深度  timestamp:服务器时间戳 | |
| ok\_btcusd\_depth60比特币60条市场深度  websocket.send("{'event':'addChannel','channel':'ok\_btcusd\_depth60'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_btcusd\_depth60'}  # Response  [{  "channel":" ok\_btcusd\_depth60",  "data":{  "bids":[[2473.88,2.025],  [2473.5,2.4],  [2470,12.203]],  "asks":[[2484,17.234],  [2483.01,6],  [2482.88,3]],  "timestamp":"1411718972024"}  }  }]    **返回值说明**  bids:买方深度  asks:卖方深度  timestamp:服务器时间戳 | |
| ok\_btcusd\_trades比特币实时成交记录  websocket.send("{'event':'addChannel','channel':'ok\_btcusd\_trades'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_btcusd\_trades'}  # Response  [{  "channel":"ok\_btcusd\_trades",  "data":{[["2463.86","0.052","16:34:07","ask"]]}  }] | |
| ok\_btcusd\_kline\_X比特币K线数据  websocket.send("{'event':'addChannel','channel':'ok\_btcusd\_kline\_X'}");  ①x值为：1min,3min,5min,15min,30min,1hour,2hour,4hour,6hour,12hour,day,3day,week②返回格式:[时间 ,开盘价,最高价,最低价,收盘价,成交量 ]  **示例**  # Request  {'event':'addChannel','channel':'ok\_btcusd\_kline\_1min'}  # Response  [{  "channel":"ok\_btcusd\_kline\_1min",  "data":{  [[1411720800000,2466.98,2467.98,2466.21,2467.97,101.3],  [1411720860000,2467.96,2467.98,2467.94,2467.98,5.89]]  }  }] | |
| ok\_ltcusd\_ticker莱特币行情数据  websocket.send("{'event':'addChannel','channel':'ok\_ltcusd\_ticker'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_ltcusd\_ticker'}  # Response  [{  "channel":" ok\_ltcusd\_ticker",  "data":{  "buy":"27.24",  "high":"28.25",  "last":"27.25",  "low":"27.13",  "sell":"27.25",  "timestamp":"1411718777857",  "vol":"6,500,950.31"  }  }]    **返回值说明**  buy: 买一价  high: 最高价  last: 最新成交价  low: 最低价  sell: 卖一价  timestamp：时间戳  vol: 成交量(最近的24小时) | |
| ok\_ltcusd\_depth莱特币20条市场深度  websocket.send("{'event':'addChannel','channel':'ok\_ltcusd\_depth'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_ltcusd\_depth'}  # Response  [{  "channel":" ok\_ltcusd\_depth",  "data":{  "bids":[[2473.88,2.025],  [23.5,2.4],  [20,12.203]],  "asks":[[24,17.234],  [23.01,6],  [22.88,3]],  "timestamp":"1411718972024"  }  }]    **返回值说明**  bids:买方深度  asks:卖方深度  timestamp:服务器时间戳 | |
| ok\_ltcusd\_depth60莱特币60条市场深度  websocket.send("{'event':'addChannel','channel':'ok\_ltcusd\_depth60'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_ltcusd\_depth60'}  # Response  [{  "channel":" ok\_ltcusd\_depth60",  "data":{  "bids":[[23.88,2.025],  [23.5,2.4],  [20,12.203]],  "asks":[[24,17.234],  [23.01,6],  [22.88,3]],  "timestamp":"1411718972024"  }  }]    **返回值说明**  bids:买方深度  asks:卖方深度  timestamp:服务器时间戳 | |
| ok\_ltcusd\_trades莱特币实时成交记录  websocket.send("{'event':'addChannel','channel':'ok\_ltcusd\_trades'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_ltcusd\_trades'}  # Response  [{  "channel":"ok\_ltcusd\_trades",  "data":{[["27.22","2.0","16:37:12","bid"]}  }] | |
| ok\_ltcusd\_kline\_X莱特币K线数据  websocket.send("{'event':'addChannel','channel':'ok\_ltcusd\_kline\_X'}");  ①x值为：1min,3min,5min,15min,30min,1hour,2hour,4hour,6hour,12hour,day,3day,week②返回格式:[时间 ,开盘价,最高价,最低价,收盘价,成交量 ]  **示例**  # Request  {'event':'addChannel','channel':'ok\_ltcusd\_kline\_1min'}  # Response  [{  "channel":"ok\_ltcusd\_kline\_1min",  "data":{[1411721040000,27.24,27.25,27.24,27.25,2903.71]}  }] | |

* **现货交易 API**
* **用于OKCoin快速进行现货交易**

|  |  |
| --- | --- |
| **接口** | **描述** |
| ok\_usd\_realtrades实时交易数据  websocket.send("{'event':'addChannel','channel':'ok\_usd\_realtrades','parameters':{'api\_key':'value1','sign':'value2'}}");  **示例**  # Request  {  'event':'addChannel',  'channel':'ok\_usd\_realtrades',  'parameters':{  'api\_key':'value1',  'sign':'value2'  }  }  # Response  [{  "channel":"ok\_usd\_realtrades",  "data":{  "averagePrice":"0",  "completedTradeAmount":"0",  "createdDate":1415945979665,  "orderId":125435440,  "status":0,  "tradeAmount":"1",  "tradePrice":"0",  "tradeType":"buy",  "tradeUnitPrice":"1",  "unTrade":"1",  }  }]  **返回值说明**  createdDate:创建日期  orderId:订单id  tradeType:交易类型：  buy:买入  sell:卖出  buy\_market:按市价买入  sell\_market:按市价卖出  tradeAmount:当按市场价买入时表示成交金额，其他表示币的成交量  tradeUnitPrice:成交单价  completedTradeAmount:已完成成交量  tradePrice:成交金额  averagePrice:平均成交价  unTrade:当按市场价买币时表示剩余金额，其他情况表示此笔交易剩余买/卖币的数量  status:0等待成交,1部分成交,2完全成交,-1撤单  **请求参数**   | **参数名** | **描述** | | --- | --- | | api\_key | 用户申请的apiKey | | sign | 请求参数的签名 | | |
| ok\_spotusd\_trade下单交易  websocket.send("{'event':'addChannel','channel':'ok\_spotusd\_trade','parameters':{'api\_key':'XXXXX','sign':'XXXX','symbol':'btc\_usd','type':'buy','price':'50','amount':'0.02'}}")  **示例**  # Request  {  'event':'addChannel',  'channel':'ok\_spotusd\_trade',  'parameters':{  'api\_key':'XXXX',  'sign':'XXXX',  'symbol':'btc\_usd',  'type':'buy',  'price':'50',  'amount':'0.02'  }  }  # Response  [{  "channel":"ok\_spotusd\_trade",  "data":{  "order\_id":"125433029",  "result":"true"  }  }]  **返回值说明**  order\_id：订单号  result：执行结果  **请求参数**   | **参数名** | **描述** | | --- | --- | | api\_key | 用户申请的apiKey | | sign | 请求参数的签名 | | symbol | btc\_usd: 比特币 ltc\_usd: 莱特币 | | type | 买卖类型： 限价单（buy/sell） 市价单（buy\_market/sell\_market） | | price | 下单价格 [限价买单(必填)： 大于等于0，小于等于1000000 | 市价买单(必填)： BTC :最少买入0.01个BTC 的金额(金额>0.01\*卖一价) / LTC :最少买入0.1个LTC 的金额(金额>0.1\*卖一价)] | | amount | 交易数量 [限价卖单（必填）：BTC 数量大于等于0.01 / LTC 数量大于等于0.1 | 市价卖单（必填）： BTC :最少卖出数量大于等于0.01 / LTC :最少卖出数量大于等于0.1] | | |
| ok\_spotusd\_cancel\_order取消订单  websocket.send("{'event':'addChannel', 'channel':'ok\_spotusd\_cancel\_order', 'parameters':{ 'api\_key':'XXXX', 'sign':'XXXX', 'symbol':'btc\_usd', 'order\_id':'125433027'} }")  **示例**  # Request  {  'event':'addChannel',  'channel':'ok\_spotusd\_cancel\_order',  'parameters':{  'api\_key':'XXXX',  'sign':'XXXX',  'symbol':'btc\_usd',  'order\_id':'125433027'}  }  # Response  [{  "channel":"ok\_spotusd\_cancel\_order",  "data":{  "order\_id":"125433027",  "result":"true"  }  }]  **返回值说明**  order\_id：订单号  result：执行结果  **请求参数**   | **参数名** | **描述** | | --- | --- | | api\_key | 用户申请的apiKey | | sign | 请求参数的签名 | | symbol | btc\_usd: 比特币 ltc\_usd: 莱特币 | | order\_id | 订单Id | | |
| ok\_spotusd\_userinfo New账户信息  websocket.send("{'event':'addChannel', 'channel':'ok\_spotusd\_userinfo', 'parameters':{ 'api\_key':'XXXX', 'sign':'XXXX'} }")  **示例**  # Request  {  'event':'addChannel',  'channel':'ok\_spotusd\_userinfo',  'parameters':{  'api\_key':'XXXX',  'sign':'XXXX'  }  }  # Response  [  {  "channel": "ok\_spotusd\_userinfo",  "data": {  "info": {  "funds": {  "asset": {  "net": "0",  "total": "0"  },  "borrow": {  "btc": "0",  "usd": "0",  "ltc": "0"  },  "free": {  "btc": "0",  "usd": "0",  "ltc": "0"  },  "freezed": {  "btc": "0",  "usd": "0",  "ltc": "0"  },  "union\_fund": {  "btc": "0",  "ltc": "0"  }  }  },  "result": true  }  }  ]  **返回值说明**  asset:账户资产，包含净资产及总资产  borrow:账户借款信息(只有在账户有借款信息时才会返回)  free:账户余额  freezed:账户冻结余额  union\_fund:账户理财信息(只有在账户有理财信息时才返回)  **请求参数**   | **参数名** | **描述** | | --- | --- | | api\_key | 用户申请的apiKey | | sign | 请求参数的签名 | | |

* **期货行情 API**
* **获取OKCoin期货行情数据**

|  |  |
| --- | --- |
| **接口** | **描述** |
| ok\_btcusd\_future\_ticker\_this\_week比特币期货当周合约行情  websocket.send("{'event':'addChannel','channel':'ok\_btcusd\_future\_ticker\_this\_week'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_btcusd\_future\_ticker\_this\_week'}  # Response  [{  "channel":"ok\_btcusd\_future\_ticker\_this\_week",  "data":{  "buy":396.93,  "contractId":20141003011,  "high":405.35,  "last":397.2,  "low":392.73,  "sell":397.6,  "unitAmount":100,  "volume":119769  }  }] | |
| ok\_btcusd\_future\_ticker\_next\_week比特币期货次周合约行情  websocket.send("{'event':'addChannel','channel':'ok\_btcusd\_future\_ticker\_next\_week'}");  **示例**  # Request  {  'event':'addChannel',  'channel':'ok\_btcusd\_future\_ticker\_next\_week'  }  # Response  [{  "channel":"ok\_btcusd\_future\_ticker\_next\_week",  "data":{  "buy":396.87,  "contractId":20141010012,  "high":405.44,  "last":397.14,  "low":392.91,  "sell":397.42,  "unitAmount":100,  "volume":36483  }  }] | |
| ok\_btcusd\_future\_ticker\_quarter比特币期货季度合约行情  websocket.send("{'event':'addChannel','channel':'ok\_btcusd\_future\_ticker\_quarter'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_btcusd\_future\_ticker\_quarter'}  # Response  [{  "channel":"ok\_btcusd\_future\_ticker\_quarter",  "data":{  "buy":397.34,  "contractId":20141226034,  "high":406.09,  "last":397.62,  "low":392.59,  "sell":398.01,  "unitAmount":100,  "volume":98288  }  }] | |
| ok\_btcusd\_kline\_this\_week\_X比特币期货当周合约K线数据  websocket.send("{'event':'addChannel','channel':'ok\_future\_btcusd\_kline\_this\_week\_X'}");  ①x值为：1min,3min,5min,15min,30min,1hour,2hour,4hour,6hour,12hour,day,3day,week②返回格式:[时间 ,开盘价,最高价,最低价,收盘价,成交量 ]  **示例**  # Request  {'event':'addChannel','channel':'ok\_future\_btcusd\_kline\_this\_week\_1min'}  # Response  [{  "channel":"ok\_future\_btcusd\_kline\_this\_week\_1min",  "data":{  [[1415160900000,331.88,332.13,331.88,332.13,55],  [1415160960000,332.13,332.13,332.13,332.13,0]]  }  }] | |
| ok\_btcusd\_kline\_next\_week\_X比特币期货次周合约K线数据  websocket.send("{'event':'addChannel','channel':'ok\_future\_btcusd\_kline\_next\_week\_X'}");  ①x值为：1min,3min,5min,15min,30min,1hour,2hour,4hour,6hour,12hour,day,3day,week②返回格式:[时间 ,开盘价,最高价,最低价,收盘价,成交量 ]  **示例**  # Request  {'event':'addChannel','channel':'ok\_future\_btcusd\_kline\_next\_week\_1min'}  # Response  [{  "channel":"ok\_future\_btcusd\_kline\_next\_week\_1min",  "data":{  [[1414988400000,330.53,330.53,330.31,330.31,43],  [1414988460000,330.31,330.31,330.31,330.31,0]]  }  }] | |
| ok\_btcusd\_kline\_quarter\_X比特币期货季度合约K线数据  websocket.send("{'event':'addChannel','channel':'ok\_future\_btcusd\_kline\_quarter\_X'}");  ①x值为：1min,3min,5min,15min,30min,1hour,2hour,4hour,6hour,12hour,day,3day,week ②返回格式:[时间 ,开盘价,最高价,最低价,收盘价,成交量 ]  **示例**  # Request  {'event':'addChannel','channel':'ok\_future\_btcusd\_kline\_quarter\_1min'}  # Response  [{  "channel":"ok\_future\_btcusd\_kline\_quarter\_1min",  "data":{  [[1414989300000,328.69,329.06,328.51,329.06,80],  [1414989360000,329.01,329.01,329.01,329.01,2]]  }  }] | |
| ok\_ltcusd\_future\_ticker\_this\_week莱特币期货当周合约行情  websocket.send("{'event':'addChannel','channel':'ok\_ltcusd\_future\_ticker\_this\_week'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_ltcusd\_future\_ticker\_this\_week'}  # Response  [{  "channel":"ok\_ltcusd\_future\_ticker\_this\_week",  "data":{  "buy":4.431,  "contractId":20141003114,  "high":4.488,  "last":4.432,  "low":4.378,  "sell":4.44,  "unitAmount":10,  "volume":119607  }  }] | |
| ok\_ltcusd\_future\_ticker\_next\_week莱特币期货次周合约行情  websocket.send("{'event':'addChannel','channel':'ok\_ltcusd\_future\_ticker\_next\_week'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_ltcusd\_future\_ticker\_next\_week'}  # Response  [{  "channel":"ok\_ltcusd\_future\_ticker\_next\_week",  "data":{  "buy":4.437,  "contractId":20141010115,  "high":4.483,  "last":4.437,  "low":4.377,  "sell":4.441,  "unitAmount":10,  "volume":90441  }  }] | |
| ok\_ltcusd\_future\_ticker\_quarter莱特币期货季度合约行情  websocket.send("{'event':'addChannel','channel':'ok\_ltcusd\_future\_ticker\_quarter'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_ltcusd\_future\_ticker\_quarter'}  # Response  [{  "channel":"ok\_ltcusd\_future\_ticker\_quarter",  "data":{  "buy":4.438,  "contractId":20141226135,  "high":4.489,  "last":4.438,  "low":4.386,  "sell":4.443,  "unitAmount":10,  "volume":100157  }  }] | |
| ok\_ltcusd\_kline\_this\_week\_X莱特币期货当周合约K线数据  websocket.send("{'event':'addChannel','channel':'ok\_future\_ltcusd\_kline\_this\_week\_X'}");  ①x值为：1min,3min,5min,15min,30min,1hour,2hour,4hour,6hour,12hour,day,3day,week ②返回格式:[时间 ,开盘价,最高价,最低价,收盘价,成交量 ]  **示例**  # Request  {'event':'addChannel','channel':'ok\_future\_ltcusd\_kline\_this\_week\_1min'}  # Response  [{  "channel":"ok\_future\_ltcusd\_kline\_this\_week\_1min",  "data":{  [[1415161080000,3.623,3.623,3.623,3.623,31],  [1415161140000,3.623,3.623,3.623,3.623,0]]  }  }] | |
| ok\_ltcusd\_kline\_next\_week\_X莱特币期货次周合约K线数据  websocket.send("{'event':'addChannel','channel':'ok\_future\_ltcusd\_kline\_next\_week\_X'}");  ①x值为：1min,3min,5min,15min,30min,1hour,2hour,4hour,6hour,12hour,day,3day,week ②返回格式:[时间 ,开盘价,最高价,最低价,收盘价,成交量 ]  **示例**  # Request  {'event':'addChannel','channel':'ok\_future\_ltcusd\_kline\_next\_week\_1min'}  # Response  [{  "channel":"ok\_future\_ltcusd\_kline\_next\_week\_1min",  "data":{  [[1414988520000,3.645,3.645,3.643,3.643,282],  [1414988580000,3.643,3.643,3.643,3.643,0]]  }  }] | |
| ok\_ltcusd\_kline\_quarter\_X莱特币期货季度合约K线数据  websocket.send("{'event':'addChannel','channel':'ok\_future\_ltcusd\_kline\_quarter\_X'}");  ①x值为：1min,3min,5min,15min,30min,1hour,2hour,4hour,6hour,12hour,day,3day,week②返回格式:[时间 ,开盘价,最高价,最低价,收盘价,成交量 ]  **示例**  # Request  {'event':'addChannel','channel':'ok\_future\_ltcusd\_kline\_quarter\_1min'}  # Response  [{  "channel":"ok\_future\_ltcusd\_kline\_quarter\_1min",  "data":{  [[1414989360000,3.658,3.658,3.658,3.658,313],  [1414989420000,3.658,3.658,3.658,3.658,0]]  }  }] | |
| ok\_btcusd\_future\_depth\_this\_week比特币期货当周市场深度  websocket.send("{'event':'addChannel','channel':'ok\_btcusd\_future\_depth\_this\_week'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_btcusd\_future\_depth\_this\_week'}  # Response  [{  "channel":"ok\_btcusd\_future\_depth\_this\_week",  "data":{  "asks":[  [398.76,4],  [398.55,20],  [398.35,14],  [397.98,98],  [397.63,4]  ],  "bids":[  [397.25,4],  [397.03,17],  [396.92,8],  [396.79,21],  [396.69,35]  ],  "timestamp":"1418636773053",  "unit\_amount":100  }  }] | |
| ok\_btcusd\_future\_depth\_next\_week比特币期货次周市场深度  websocket.send("{'event':'addChannel','channel':'ok\_btcusd\_future\_depth\_next\_week'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_btcusd\_future\_depth\_next\_week'}  # Response  [{  "channel":"ok\_btcusd\_future\_depth\_next\_week",  "data":{  "asks":[  [398.32,35],  [398.19,20],  [398,87],  [397.99,22],  [397.74,32]  ],  "bids":[  [397.3,9],  [397,7],  [396.9,20],  [396.78,5],  [396.66,13]  ],  "timestamp":"1418636773053",  "unit\_amount":100  }  }] | |
| ok\_btcusd\_future\_depth\_quarter比特币期货季度市场深度  websocket.send("{'event':'addChannel','channel':'ok\_btcusd\_future\_depth\_quarter'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_btcusd\_future\_depth\_quarter'}  # Response  [{  "channel":"ok\_btcusd\_future\_depth\_quarter",  "data":{  "asks":[  [398.32,35],  [398.19,20],  [398,87],  [397.99,22],  [397.74,32]  ],  "bids":[  [397.3,9],  [397,7],  [396.9,20],  [396.78,5],  [396.66,13]  ],  "timestamp":"1418636773053",  "unit\_amount":100  }  }] | |
| ok\_ltcusd\_future\_depth\_this\_week莱特币期货当周市场深度  websocket.send("{'event':'addChannel','channel':'ok\_ltcusd\_future\_depth\_this\_week'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_ltcusd\_future\_depth\_this\_week'}  # Response  [{  "channel":"ok\_ltcusd\_future\_depth\_this\_week",  "data":{  "asks":[  [4.444,247],  [4.443,90],  [4.441,206],  [4.44,141],  [4.439,142]  ],  "bids":[  [4.433,298],  [4.429,78],  [4.428,230],  [4.426,124],  [4.425,109]  ],  "timestamp":"1418636773053",  "unit\_amount":10  }  }] | |
| ok\_ltcusd\_future\_depth\_next\_week莱特币期货次周市场深度  websocket.send("{'event':'addChannel','channel':'ok\_ltcusd\_future\_depth\_next\_week'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_ltcusd\_future\_depth\_next\_week'}  # Response  [{  "channel":"ok\_ltcusd\_future\_depth\_next\_week",  "data":{  "asks":[  [4.445,243],  [4.444,276],  [4.441,94],  [4.439,234],  [4.437,41]  ],  "bids":[  [4.431,153],  [4.429,73],  [4.428,34],  [4.425,145],  [4.424,37]  ],  "timestamp":"1418636773053",  "unit\_amount":10  }  }] | |
| ok\_ltcusd\_future\_depth\_quarter莱特币期货季度市场深度  websocket.send("{'event':'addChannel','channel':'ok\_ltcusd\_future\_depth\_quarter'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_ltcusd\_future\_depth\_quarter'}  # Response  [{  "channel":"ok\_ltcusd\_future\_depth\_quarter",  "data":{  "asks":[  [4.45,50],  [4.449,104],  [4.446,57],  [4.444,251],  [4.434,18]  ],  "bids":[  [4.433,120],  [4.431,35],  [4.429,134],  [4.426,236],  [4.424,180]  ],  "timestamp":"1418636773053",  "unit\_amount":10  }  }] | |
| ok\_btcusd\_future\_trade\_this\_week比特币期货当周交易信息  websocket.send("{'event':'addChannel','channel':'ok\_btcusd\_future\_trade\_this\_week'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_btcusd\_future\_trade\_this\_week'}  # Response  [{  "channel":"ok\_btcusd\_future\_trade\_this\_week",  "data":{[  ["397.04","8.0","12:42:48","ask"],  ["397.0","6.0","12:44:43","bid"],  ["397.01","6.0","12:45:36","bid"]  ]  }] | |
| ok\_btcusd\_future\_trade\_next\_week比特币期货次周交易信息  websocket.send("{'event':'addChannel','channel':'ok\_btcusd\_future\_trade\_next\_week'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_btcusd\_future\_trade\_next\_week'}  # Response  [{  "channel":"ok\_btcusd\_future\_trade\_next\_week",  "data":{  [["396.65","10.0","12:43:37","ask"],  ["396.72","34.0","12:45:27","ask"],  ["396.73","2.0","12:46:46","ask"]  ]  }  }] | |
| ok\_btcusd\_future\_trade\_quarter比特币期货季度交易信息  websocket.send("{'event':'addChannel','channel':'ok\_btcusd\_future\_trade\_quarter'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_btcusd\_future\_trade\_quarter'}  # Response  [{  "channel":"ok\_btcusd\_future\_trade\_quarter",  "data":{  [["396.36","8.0","12:43:32","ask"],  ["396.42","168.0","12:44:43","ask"],  ["396.43","4.0","12:45:14","ask"]  ]  }  }] | |
| ok\_ltcusd\_future\_trade\_this\_week莱特币期货当周交易信息  websocket.send("{'event':'addChannel','channel':'ok\_ltcusd\_future\_trade\_this\_week'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_ltcusd\_future\_trade\_this\_week'}  # Response  [{  "channel":"ok\_ltcusd\_future\_trade\_this\_week",  "data":{  [["4.425","415.0","12:44:46","ask"],  ["4.428","1.0","12:44:52","ask"],  ["4.438","16.0","12:46:38","ask"]  ]  }  }] | |
| ok\_ltcusd\_future\_trade\_next\_week莱特币期货次周交易信息  websocket.send("{'event':'addChannel','channel':'ok\_ltcusd\_future\_trade\_next\_week'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_ltcusd\_future\_trade\_next\_week'}  # Response  [{  "channel":"ok\_ltcusd\_future\_trade\_next\_week",  "data":{  [["4.432","8.0","12:56:28","ask"],  ["4.434","56.0","12:57:39","ask"],  ["4.432","98.0","12:59:10","ask"]  ]  }  }] | |
| ok\_ltcusd\_future\_trade\_quarter莱特币期货季度交易信息  websocket.send("{'event':'addChannel','channel':'ok\_ltcusd\_future\_trade\_quarter'}");  **示例**  # Request  {'event':'addChannel','channel':'ok\_ltcusd\_future\_trade\_quarter'}  # Response  [{  "channel":"ok\_ltcusd\_future\_trade\_quarter",  "data":{  [["4.435","68.0","12:56:36","ask"],  ["4.435","50.0","12:58:01","ask"],  ["4.434","64.0","12:59:17","ask"]]  }  }] | |

* **期货交易 API**
* **获取OKCoin期货交易数据**

|  |  |
| --- | --- |
| **接口** | **描述** |
| ok\_futuresusd\_trade New期货下单交易  websocket.send("{'event': 'addChannel','channel':'ok\_futuresusd\_trade','parameters': {'api\_key':'XXXXX','sign':'XXXXX','symbol': 'btc\_usd','contract\_type': 'this\_week','price': '356.5','amount': '2','type': '1','match\_price': '0','lever\_rate': '20'}}")  **示例**  # Request  {  'event': 'addChannel',  'channel': 'ok\_futuresusd\_trade',  'parameters': {  'api\_key': 'XXXXX',  'sign': 'XXXXX',  'symbol': 'btc\_usd',  'contract\_type': 'this\_week',  'price': '356.5',  'amount': '2',  'type': '1',  'match\_price': '0',  'lever\_rate': '20'  }  }  # Response  [  {  "channel": "ok\_futuresusd\_trade",  "data": {  "order\_id": "125433029",  "result": "true"  }  }  ]  **返回值说明**  order\_id: 订单号  result: 执行结果  **请求参数**   | **参数名** | **描述** | | --- | --- | | api\_key | 用户申请的apiKey | | sign | 请求参数的签名 | | symbol | btc\_usd: 比特币 ltc\_usd: 莱特币 | | contract\_type | 合约类型: this\_week:当周 next\_week:下周 quarter:季度 | | price | 价格 | | amount | 委托数量 | | type | 1:开多 2:开空 3:平多 4:平空 | | match\_price | 是否为对手价： 0:不是 1:是 当取值为1时,price无效 | | lever\_rate | 杠杆倍数 value:10\20 默认10 | | |
| ok\_futuresusd\_cancel\_order New取消期货订单  websocket.send("{'event': 'addChannel','channel': 'ok\_futuresusd\_cancel\_order','parameters': {'api\_key':'XXXXX','sign':'XXXXX','symbol': 'btc\_usd','order\_id': '125433027','contract\_type': 'this\_week'}}")  **示例**  # Request  {  'event': 'addChannel',  'channel': 'ok\_futuresusd\_cancel\_order',  'parameters': {  'api\_key': 'XXXXX',  'sign':'XXXXX',  'symbol': 'btc\_usd',  'order\_id': '125433027',  'contract\_type': 'this\_week'  }  }  # Response  [  {  "channel": "ok\_futuresusd\_cancel\_order",  "data": {  "order\_id": "125433029",  "result": "true"  }  }  ]  **返回值说明**  order\_id: 订单号  result: 执行结果  **请求参数**   | **参数名** | **描述** | | --- | --- | | api\_key | 用户申请的apiKey | | sign | 请求参数的签名 | | symbol | btc\_usd: 比特币 ltc\_usd: 莱特币 | | order\_id | 订单ID | | contract\_type | 合约类型: this\_week:当周 next\_week:下周 quarter:季度 | | |
| ok\_usd\_future\_realtrades New实时交易数据  websocket.send("{'event':'addChannel','channel':'ok\_usd\_future\_realtrades','parameters':{'api\_key':'value1','sign':'value2'}}");  **示例**  # Request  {  'event':'addChannel',  'channel':'ok\_usd\_future\_realtrades',  'parameters':  {  'api\_key':'value1',  'sign':'value2'  }  }  # Response  [{  "channel":"ok\_usd\_future\_realtrades",  "data":{  "amount":"1",  "contract\_id":"20141031011",  "contract\_name":"BTC1031",  "contract\_type":"this\_week",  "deal\_amount":"0",  "fee":"0",  "orderid":161705,  "price":"1",  "price\_avg":"0",  "status":"0",  "type":"1",  "unit\_amount":"100",  "lever\_rate":"20"  }  }]  **返回值说明**  amount : 委托量  contract\_id : 合约  contract\_name : 合约名称  contract\_type : 合约类型  deal\_amount : 成交数量  fee : 手续费  orderid : 订单ID  price : 委托价格  price\_avg : 平均价格  status : 0等待成交 1部分成交 2全部成交 -1撤单  type : 1.开多 2.开空 3.平多 4.平空  unit\_amount : 合约面值  lever\_rate: 杠杆倍数  **请求参数**   | **参数名** | **描述** | | --- | --- | | api\_key | 用户申请的apiKey | | sign | 请求参数的签名 | | |