



现货/杠杆/币安宝/矿池 U本位合约 币本位合约 欧式期权

[English](#)

- [更新日志](#)
- [介绍](#)
 - [API Key 设置](#)
 - [API Key 权限设置](#)
 - [账户](#)
 - [API 代码库](#)
 - [联系我们](#)
- [基本信息](#)
 - [API 基本信息](#)
 - [访问限制](#)
 - [数据来源](#)
 - [接口鉴权类型](#)
 - [SIGNED \(TRADE、USER DATA AND MARGIN\) Endpoint security.](#)
 - [公开 API 参数](#)
 - [过滤器](#)
- [钱包接口](#)
 - [系统状态 \(System\)](#)
 - [获取所有币信息 \(USER DATA\)](#)
 - [查询每日资产快照 \(USER DATA\)](#)
 - [关闭站内划转 \(USER DATA\)](#)
 - [开启站内划转 \(USER DATA\)](#)
 - [提币 \(USER DATA\)](#)
 - [获取充值历史 \(支持多网络\) \(USER DATA\)](#)
 - [获取提币历史 \(支持多网络\) \(USER DATA\)](#)
 - [获取充值地址 \(支持多网络\) \(USER DATA\)](#)
 - [账户状态 \(USER DATA\)](#)
 - [账户 API 交易状态 \(USER DATA\)](#)
 - [小额资产转换 BNB 历史 \(USER DATA\)](#)
 - [获取可以转换成 BNB 的小额资产 \(USER DATA\)](#)
 - [小额资产转换 \(USER DATA\)](#)
 - [资产利息记录 \(USER DATA\)](#)
 - [上架资产详情 \(USER DATA\)](#)
 - [交易手续费率查询 \(USER DATA\)](#)
 - [用户万向划转 \(USER DATA\)](#)
 - [查询用户万向划转历史 \(USER DATA\)](#)
 - [资金账户 \(USER DATA\)](#)
 - [用户持仓 \(USER DATA\)](#)
 - [稳定币自动兑换划转 \(TRADE\)](#)
 - [稳定币自动兑换划转查询 \(USER DATA\)](#)
 - [查询用户 API Key 权限 \(USER DATA\)](#)
- [子母账户接口](#)
 - [创建虚拟子账户 \(适用主账户\)](#)
 - [查询子账户列表 \(适用主账户\)](#)
 - [查询子账户现货资金划转历史 \(适用主账户\)](#)
 - [查询子账户合约资金划转历史 \(适用主账户\)](#)
 - [执行子账户合约资金划转 \(适用主账户\)](#)
 - [查询子账户资产 \(适用主账户\)](#)
 - [查询子账户现货资产汇总 \(适用主账户\)](#)

- [获取子账户充值地址 \(适用主账户\)](#)
- [获取子账户充值记录 \(适用主账户\)](#)
- [查询子账户Margin/Futures状态 \(适用主账户\)](#)
- [为子账户开通Margin \(适用主账户\)](#)
- [查询子账户Margin账户详情 \(适用主账户\)](#)
- [查询子账户Margin账户汇总 \(适用主账户\)](#)
- [为子账户开通Futures \(适用主账户\)](#)
- [查询子账户Futures账户详情 \(适用主账户\)](#)
- [查询子账户Futures账户汇总 \(适用主账户\)](#)
- [查询子账户合约持仓信息 \(仅适用主账户\)](#)
- [子账户Futures划转 \(仅适用主账户\)](#)
- [子账户Margin划转 \(仅适用主账户\)](#)
- [向共同主账户下的子账户主动划转 \(仅适用于子账户\)](#)
- [向主账户主动划转 \(仅适用于子账户\)](#)
- [查询子账户划转历史 \(仅适用于子账户\)](#)
- [子母账户万能划转 \(适用主账户\)](#)
- [查询子母账户万能划转历史 \(适用主账户\)](#)
- [查询子账户Futures账户详情V2 \(适用主账户\)](#)
- [查询子账户Futures账户汇总V2 \(适用主账户\)](#)
- [查询子账户合约持仓信息V2 \(仅适用主账户\)](#)
- [为子账户开通杠杆代币 \(适用母账户\)](#)
- [为子账户API Key开启/关闭IP白名单 \(适用母账户\)](#)
- [为子账户API Key添加IP白名单 \(适用母账户\)](#)
- [查询子账户API Key IP白名单 \(适用母账户\)](#)
- [删除子账户API Key IP白名单 \(适用母账户\)](#)
- [投资人账户为托管子账户充值资产 \(适用投资人母账户\)](#)
- [投资人账户查询托管子账户资产 \(适用投资人母账户\)](#)
- [投资人账户为托管子账户提币资产 \(适用投资人母账户\)](#)
- [查询托管子账户资产快照 \(适用投资人母账户\)](#)

- [行情接口](#)

- [测试服务器连通性](#)
- [获取服务器时间](#)
- [交易规范信息](#)
- [深度信息](#)
- [近期成交列表](#)
- [查询历史成交 \(MARKET DATA\)](#)
- [近期成交\(归集\)](#)
- [K线数据](#)
- [当前平均价格](#)
- [UIK线数据](#)
- [24hr 价格变动情况](#)
- [最新价格](#)
- [当前最优挂单](#)
- [滚动窗口价格变动统计](#)

- [WebSocket 行情推送](#)

- [实时订阅/取消数据流](#)
- [归集交易流](#)
- [逐笔交易](#)
- [K线 Streams](#)
- [按 Symbol 的精简Ticker](#)
- [全市场所有Symbol的精简Ticker](#)
- [按Symbol的完整Ticker](#)
- [全市场所有交易对的完整Ticker](#)
- [按Symbol的最优挂单信息](#)
- [全市场最优挂单信息](#)

- [有限档深度信息](#)
- [增量深度信息](#)
- [按Symbol的滚动窗口统计](#)
- [全市场滚动窗口统计](#)
- [如何正确在本地维护一个orderbook副本](#)
- [现货账户和交易接口](#)
 - [测试下单 \(TRADE\)](#)
 - [下单 \(TRADE\)](#)
 - [撤销订单 \(TRADE\)](#)
 - [撤销单一交易对的所有挂单 \(TRADE\)](#)
 - [撤消挂单再下单 \(TRADE\)](#)
 - [查询订单 \(USER DATA\)](#)
 - [当前挂单 \(USER DATA\)](#)
 - [查询所有订单 \(USER DATA\)](#)
 - [OCO下单\(TRADE\)](#)
 - [取消 OCO 订单\(TRADE\)](#)
 - [查询 OCO \(USER DATA\)](#)
 - [查询所有 OCO \(USER DATA\)](#)
 - [查询 OCO 挂单 \(USER DATA\)](#)
 - [账户信息 \(USER DATA\)](#)
 - [账户成交历史 \(USER DATA\)](#)
 - [查询目前下单数 \(TRADE\)](#)
- [杠杆账户和交易接口](#)
 - [全仓杠杆账户划转 \(MARGIN\)](#)
 - [杠杆账户借贷 \(MARGIN\)](#)
 - [杠杆账户归还借贷 \(MARGIN\)](#)
 - [查询杠杆资产 \(MARKET DATA\)](#)
 - [查询全仓杠杆交易对 \(MARKET DATA\)](#)
 - [获取所有杠杆资产信息 \(MARKET DATA\)](#)
 - [获取所有全仓杠杆交易对\(MARKET DATA\)](#)
 - [查询杠杆价格指数 \(MARKET DATA\)](#)
 - [杠杆账户下单 \(TRADE\)](#)
 - [杠杆账户撤销订单 \(TRADE\)](#)
 - [杠杆账户撤销单一交易对的所有挂单 \(TRADE\)](#)
 - [获取全仓杠杆划转历史 \(USER DATA\)](#)
 - [查询借贷记录 \(USER DATA\)](#)
 - [查询还贷记录 \(USER DATA\)](#)
 - [获取利息历史 \(USER DATA\)](#)
 - [获取账户强制平仓记录\(USER DATA\)](#)
 - [查询全仓杠杆账户详情 \(USER DATA\)](#)
 - [查询杠杆账户订单 \(USER DATA\)](#)
 - [查询杠杆账户挂单记录 \(USER DATA\)](#)
 - [查询杠杆账户的所有订单 \(USER DATA\)](#)
 - [杠杆账户 OCO 下单\(TRADE\)](#)
 - [取消杠杆账户 OCO 订单\(TRADE\)](#)
 - [查询杠杆账户 OCO \(USER DATA\)](#)
 - [查询特定杠杆账户所有 OCO \(USER DATA\)](#)
 - [查询杠杆账户 OCO 挂单 \(USER DATA\)](#)
 - [查询杠杆账户交易历史 \(USER DATA\)](#)
 - [查询账户最大可借贷额度\(USER DATA\)](#)
 - [查询最大可转出额 \(USER DATA\)](#)
 - [查询Margin账户信息汇总 \(USER DATA\)](#)
 - [杠杆逐仓账户划转 \(MARGIN\)](#)
 - [获取杠杆逐仓划转历史 \(USER DATA\)](#)
 - [查询杠杆逐仓账户信息 \(USER DATA\)](#)

- [杠杆逐仓账户停用 \(TRADE\)](#)
- [杠杆逐仓账户启用 \(TRADE\)](#)
- [查询杠杆逐仓账户启用限制 \(USER DATA\)](#)
- [查询逐仓杠杆交易对 \(USER DATA\)](#)
- [获取所有逐仓杠杆交易对 \(USER DATA\)](#)
- [现货交易和杠杆利息BNB抵扣开关 \(USER DATA\)](#)
- [获取BNB抵扣开关状态 \(USER DATA\)](#)
- [获取杠杆利率历史 \(USER DATA\)](#)
- [获取全仓杠杆利率及限额 \(USER DATA\)](#)
- [获取逐仓杠杆利率及限额 \(USER DATA\)](#)
- [获取逐仓档位信息 \(USER DATA\)](#)
- [查询目前杠杆账户下单数 \(TRADE\)](#)
- [杠杆小额资产转换BNB历史 \(USER DATA\)](#)
- [WebSocket账户信息推送](#)
 - [Listen Key\(现货账户\)](#)
 - [Listen Key\(杠杆账户\)](#)
 - [Listen Key\(逐仓杠杆账户\)](#)
 - [Payload: 账户更新](#)
 - [Payload: 余额更新](#)
 - [Payload: 订单更新](#)
- [币安宝接口](#)
 - [获取活期产品列表 \(USER DATA\)](#)
 - [获取用户当日剩余活期可申购余额 \(USER DATA\)](#)
 - [申购活期产品 \(USER DATA\)](#)
 - [获取用户当日活期可赎回余额 \(USER DATA\)](#)
 - [赎回活期产品 \(USER DATA\)](#)
 - [用户活期产品持仓 \(USER DATA\)](#)
 - [查询定期/活动产品列表 \(USER DATA\)](#)
 - [申购定期/活动产品 \(USER DATA\)](#)
 - [用户定期/活动持仓 \(USER DATA\)](#)
 - [币安宝账户信息 \(USER DATA\)](#)
 - [获取申购记录 \(USER DATA\)](#)
 - [获取赎回记录 \(USER DATA\)](#)
 - [获取利息历史 \(USER DATA\)](#)
 - [定期/活动持仓转活期持仓 \(USER DATA\)](#)
- [Staking 接口](#)
 - [查询Staking产品列表 \(USER DATA\)](#)
 - [申购锁仓产品 \(USER DATA\)](#)
 - [赎回锁仓产品 \(USER DATA\)](#)
 - [查看个人持仓 \(USER DATA\)](#)
 - [查看Staking历史记录 \(USER DATA\)](#)
 - [设置自动续期 \(USER DATA\)](#)
 - [查询Staking个人剩余额度 \(USER DATA\)](#)
- [矿池接口](#)
 - [获取算法 \(MARKET DATA\)](#)
 - [获取币种 \(MARKET DATA\)](#)
 - [请求矿工列表明细 \(USER DATA\)](#)
 - [请求矿工列表 \(USER DATA\)](#)
 - [收益列表 \(USER DATA\)](#)
 - [其他收益列表 \(USER DATA\)](#)
 - [算力转让详情列表 \(USER DATA\)](#)
 - [算力转让列表 \(USER DATA\)](#)
 - [算力转让详情 \(USER DATA\)](#)
 - [算力转让请求 \(USER DATA\)](#)
 - [取消算力转让设置 \(USER DATA\)](#)

- [统计列表 \(USER DATA\)](#)
- [账号列表 \(USER DATA\)](#)
- [矿池账户收益列表 \(USER DATA\)](#)
- [合约接口](#)
 - [合约资金划转 \(USER DATA\)](#)
 - [获取合约资金划转历史 \(USER DATA\)](#)
 - [混合保证金借款历史 \(USER DATA\)](#)
 - [混合保证金还款历史 \(USER DATA\)](#)
 - [混合保证金钱包V2 \(USER DATA\)](#)
 - [混合保证金调整质押率历史 \(USER DATA\)](#)
 - [混合保证金强平历史 \(USER DATA\)](#)
 - [混合保证金利息收取历史 \(USER DATA\)](#)
- [合约策略交易接口](#)
 - [成交量份额参与算法\(VP\)下单 \(TRADE\)](#)
 - [时间加权平均价格策略\(Twap\)下单 \(TRADE\)](#)
 - [取消策略订单 \(TRADE\)](#)
 - [查询当前策略订单挂单 \(USER DATA\)](#)
 - [查询历史策略订单 \(USER DATA\)](#)
 - [查询执行子订单 \(USER DATA\)](#)
- [统一账户接口](#)
 - [查询统一账户信息 \(USER DATA\)](#)
 - [统一账户资产质押率 \(MARKET DATA\)](#)
 - [查询统一账户穿仓借贷金额 \(USER DATA\)](#)
 - [偿还统一账户穿仓负债](#)
- [杠杆代币接口](#)
 - [杠杆代币信息 \(MARKET DATA\)](#)
 - [杠杆代币历史净值K线](#)
 - [申购代币 \(USER DATA\)](#)
 - [查询申购记录 \(USER DATA\)](#)
 - [赎回代币 \(USER DATA\)](#)
 - [查询赎回记录 \(USER DATA\)](#)
 - [查询用户每日申购赎回限额 \(USER DATA\)](#)
 - [Websocket 杠杆代币信息更新](#)
 - [Websocket 杠杆代币净值K线更新](#)
- [币安挖矿接口](#)
 - [获取所有流动资金池 \(MARKET DATA\)](#)
 - [获取流动资金池具体信息 \(USER DATA\)](#)
 - [添加流动性 \(TRADE\)](#)
 - [移除流动性 \(TRADE\)](#)
 - [获取流动性操作记录 \(USER DATA\)](#)
 - [获取报价 \(USER DATA\)](#)
 - [交易 \(TRADE\)](#)
 - [获取交易记录 \(USER DATA\)](#)
 - [获取币对池的配置信息 \(USER DATA\)](#)
 - [添加流动性的试算 \(USER DATA\)](#)
 - [移除流动性的试算 \(USER DATA\)](#)
 - [查询未领取的奖励数量 \(USER DATA\)](#)
 - [领取奖励 \(TRADE\)](#)
 - [获取已领取奖励记录 \(USER DATA\)](#)
- [法币接口](#)
 - [获取法币充值/提现历史记录 \(USER DATA\)](#)
 - [获取法币支付历史记录 \(USER DATA\)](#)
- [C2C 接口](#)
 - [获取 C2C 交易历史记录 \(USER DATA\)](#)
- [质押借币接口](#)

- [获取质押借币资金流水 \(USER DATA\)](#)
- [借币 - 质押借币借贷 \(TRADE\)](#)
- [借币 - 查询质押借币历史记录 \(USER DATA\)](#)
- [借币 - 查询借款中订单列表 \(USER DATA\)](#)
- [还款 - 质押借币还款 \(TRADE\)](#)
- [还款 - 查询还款记录历史 \(USER DATA\)](#)
- [调整质押率 - 质押借币调整质押率 \(TRADE\)](#)
- [调整质押率 - 查询质押率调整历史 \(USER DATA\)](#)
- [查询可借币种数据 \(USER DATA\)](#)
- [查询抵押币种数据 \(USER DATA\)](#)
- [查询抵押币种还款汇率 \(USER DATA\)](#)
- [质押借币自定义补仓质押率 \(TRADE\)](#)
- [Pay 接口](#)
 - [获取 Pay 交易历史记录 \(USER DATA\)](#)
- [闪兑接口](#)
 - [获取闪兑交易记录 \(USER DATA\)](#)
- [返佣接口](#)
 - [获取现货返佣历史记录 \(USER DATA\)](#)
- [NFT 接口](#)
 - [获取 NFT 资金流水记录 \(USER DATA\)](#)
 - [获取 NFT 充值记录 \(USER DATA\)](#)
 - [获取 NFT 提现记录 \(USER DATA\)](#)
 - [获取 NFT 资产 \(USER DATA\)](#)
- [币安码接口](#)
 - [创建币安码 \(USER DATA\)](#)
 - [兑现币安码 \(USER DATA\)](#)
 - [验证币安码 \(USER DATA\)](#)
 - [获取RSA Public Key \(USER DATA\)](#)
 - [购买币安码 \(TRADE\)](#)
 - [获取货币使用限制 \(USER DATA\)](#)
- [错误代码](#)
 - [10xx - 常规服务器或网络问题](#)
 - [11xx - 2xxx Request issues](#)
 - [3xxx-5xxx SAPI 具体问题](#)
 - [6XXX - 币安宝相关](#)
 - [70xx - 期货](#)
 - [20xxx - 期货策略交易](#)
 - [-9xxx 过滤器故障](#)
 - [10xxx - 质押借币](#)
 - [13xxx - 杠杆代币](#)
 - [12xxx - 流动性挖矿](#)
 - [18xxx - 币安码](#)
 - [21xxx - 统一账户](#)
 - [订单拒绝错误](#)
 - [关于 POST /api/v3/order/cancelReplace 的错误](#)
- [备注说明](#)
 - [请求参数](#)

更新日志

2022-11-02

- 更新钱包接口 :

- POST /sapi/v1/capital/withdraw/apply : 权重改至 Weight(UID) 600。

2022-11-01

- 添加质押借币接口 :

- GET /sapi/v1/loan/loanable/data : 获取可借币种的利率和借贷限额。借入限额以美元价值显示。
- GET /sapi/v1/loan/collateral/data : 获取抵押币种质押率信息和质押限额。质押限额以美元价值显示。
- GET /sapi/v1/loan/repay/collateral/rate : 获取抵押物还款时，抵押/借货币种的汇率价格。汇率价格有效时间为8秒。
- POST /sapi/v1/loan/customize/margin_call : 质押借币自定义补仓质押率，仅可针对进行中订单，自定义补仓质押率。

2022-10-28

- 更新钱包接口 :

- POST /sapi/v1/asset/convert-transfer : 增加 accountType 参数
- POST /sapi/v1/asset/convert-transfer/queryByPage : 改为 GET 请求方式，增加 clientTranId 参数

2022-10-15

- 添加币安码接口 :

- POST /sapi/v1/giftcard/buyCode : 用于购买一个币安码
- GET /sapi/v1/giftcard/buyCode/token-limit : 用来查看你所支付的数字货币，可以购买的面额与数量限制

2022-09-30

- 删除合约混合保证金接口 :

- POST /sapi/v1/futures/loan/borrow
- POST /sapi/v1/futures/loan/repay
- GET /sapi/v1/futures/loan/configs
- GET /sapi/v2/futures/loan/configs
- GET /sapi/v1/futures/loan/calcAdjustLevel
- GET /sapi/v2/futures/loan/calcAdjustLevel
- GET /sapi/v1/futures/loan/calcMaxAdjustAmount
- GET /sapi/v2/futures/loan/calcMaxAdjustAmount
- POST /sapi/v1/futures/loan/adjustCollateral
- POST /sapi/v2/futures/loan/adjustCollateral
- GET /sapi/v1/futures/loan/collateralRepayLimit
- GET /sapi/v1/futures/loan/collateralRepay
- POST /sapi/v1/futures/loan/collateralRepay
- GET /sapi/v1/futures/loan/collateralRepayResult

2022-09-30

`!bookTicker` 的 WebSocket 推送的变更.

- 全市场最优挂单信息推送(`!bookTicker`)计划在**2022年11月**下线, 具体下线的时间会在后面通告.
- 请使用按Symbol的最优挂单信息推送(`<symbol>@bookTicker`).
- 多个 `<symbol>@bookTicker` 可以订阅在一个WebSocket连接上.
 - 比如 `wss://stream.binance.com:9443/stream?streams=btcusdt@bookTicker/bnbbtc@bookTicker`

2022-09-29

- 添加钱包接口 :

- POST /sapi/v1/asset/convert-transfer: 稳定币自动兑换划转
- POST /sapi/v1/asset/convert-transfer/queryByPage: 稳定币自动兑换划转查询

2022-09-22

- 更新子母账户接口：
 - POST /sapi/v1/sub-account/subAccountApi/ipRestriction: 添加一个新的参数 thirdParty
 - POST /sapi/v1/sub-account/subAccountApi/ipRestriction/ipList: 添加一个新的参数 thirdPartyName
 - DELETE /sapi/v1/sub-account/subAccountApi/ipRestriction/ipList: 添加一个新的参数 thirdPartyName
- 添加频次限制：
 - GET /sapi/v1/bswap/liquidity: 每个账户每个池子最多每秒三次
 - GET /sapi/v1/bswap/quote: 每个账户最多三秒一次
 - POST /sapi/v1/lending/daily/purchase: 每个账户最多三秒一次
 - POST /sapi/v1/lending/customizedFixed/purchase: 每个账户最多三秒一次
 - POST /sapi/v1/staking/purchase: 每个账户最多三秒一次

2022-09-16

- 添加杠杆账户接口：
 - GET /sapi/v1/margin/tradeCoeff: 获取用户个人杠杆账户信息汇总

2022-09-15

- 添加质押借币接口：
 - POST /sapi/v1/loan/borrow: 借币 - 质押借币借贷
 - GET /sapi/v1/loan/borrow/history: 借币 - 查询质押借币历史记录
 - GET /sapi/v1/loan/ongoing/orders: 借币 - 查询借款中订单列表
 - POST /sapi/v1/loan/repay: 还款 - 质押借币还款
 - GET /sapi/v1/loan/repay/history: 还款 - 查询还款记录历史
 - POST /sapi/v1/loan/adjust/ltv: 调整质押率 - 质押借币调整质押率
 - GET /sapi/v1/loan/ltv/adjustment/history: 调整质押率 - 查询质押率调整历史

2022-09-15

这些变动会是滚动发布，可能需要几天才会部署到所有服务器。

- 接口 GET /api/v3/exchangeInfo 的变动
 - 添加一个新的参数 permissions, 用于查询适用于相应权限的所有交易对.
 - 如果查询时不提供此参数, 则默认值是 ["SPOT", "MARGIN", "LEVERAGED"].
 - 这表示如果请求 GET /api/v3/exchangeInfo 时候没有任何参数, 则会返回拥有权限是 SPOT, MARGIN, LEVERAGED 的交易对.
 - 如果要查询其他交易权限, 比如 TRD_GRP_004 等, 需要在查询参数里设置(比如 permissions = TRD_GRP_004).
 - 此参数不可以同时和 symbol 或者 symbols 使用.

2022-09-12

- 更新子母账户接口：
 - GET /sapi/v1/sub-account/subAccountApi/ipRestriction: 以支持母账户为子账户API Key查询三方IP白名单

2022-09-05

- 删除期货接口：
 - GET /sapi/v1/futures/loan/wallet

2022-08-23

这些变动会是滚动发布，可能需要几天才会部署到所有服务器。

- 接口 `GET /api/v3/ticker` 与 `GET /api/v3/ticker/24hr` 变动
 - 添加新可选参数 `type`
 - `type` 可接受的参数值有 `FULL` 与 `MINI`
 - `FULL` 是默认值，也是原来接口所返回的响应
 - `MINI` 省略了以下字段: `priceChangePercent`, `weightedAvgPrice`, `bidPrice`, `bidQty`, `askPrice`, `askQty` 与 `lastQty`
- 添加新错误代码 `-1008`
 - 每当服务器的请求超载时都会发送此消息
 - 此错误代码只会在 SPOT API 里出现
- 接口 `GET /api/v3/account` 添加新参数 `brokered`
- 添加新接口: `GET /api/v3/uiKlines`
- 添加新K线间隔: `1s`

2022-08-18

- 更新闪兑接口：
 - `GET /sapi/v1/convert/tradeFlow`: 权重自 Weight(IP) 3000改至 Weight(UID) 3000。

2022-08-08

REST API

- 接口 `POST /api/v3/order` 与 `POST /api/v3/order/cancelReplace` 变动
 - 添加新可选参数 `strategyId` 是用于将订单标识为某策略的参数。
 - 添加新可选参数 `strategyType` 是用于标识在执行的策略。(例如：如果所有订单属于现货网格策略，订单可设置为 `strategyType=1000000`)
- 接口 `POST /api/v3/order/oco` 变动
 - 添加新可选参数 `limitStrategyId`, `limitStrategyType`, `stopStrategyId`, `stopStrategyType`
 - 这些是OCO订单里两个leg的策略元数据
 - `limitStrategyType` 和 `stopStrategyType` 都不能低于 `1000000`
- 接口 `GET /api/v3/order`, `GET /api/v3/openOrders` 与 `GET /api/v3/allOrders` 变动
 - 新增参数 `strategyId` 与 `strategyType` 必须在下单时填上字段才会在回应JSON里返回
- 接口 `DELETE /api/v3/order` 与 `DELETE /api/v3/openOrders` 变动
 - 新增参数 `strategyId` 与 `strategyType` 必须在下单时填上字段才会在回应JSON里返回

USER DATA STREAM

- `eventType executionReport` 新增参数
 - `j` 代表 `strategyId`
 - `J` 代表 `strategyType`
 - 必须在下单时填上字段才会在回应里返回

2022-08-05

- 更新闪兑接口：
 - `GET /sapi/v1/convert/tradeFlow`: 权重自 Weight(IP) 100改至 Weight(IP) 3000。

2022-07-21

- 添加新统一账户接口：
 - `GET /sapi/v1/portfolio/pmLoan` 查询统一账户穿仓借贷记录。
 - `POST /sapi/v1/portfolio/repay` 偿还统一账户穿仓负债。

2022-07-18

- 添加新统一账户接口：
 - GET /sapi/v1/portfolio/collateralRate 获取统一账户资产质押率。

2022-07-01

- 添加新钱包接口：
 - POST /sapi/v3/asset/getUserAsset 获取用户持仓。
- 添加新杠杆账户接口：
 - GET /sapi/v1/margin/dribblet 查询用户杠杆账户小额资产转换BNB历史信息。
- 更新闪兑接口：
 - GET /sapi/v1/convert/tradeFlow：权重自3000改至100。
- 更新杠杆账户接口：
 - GET /sapi/v1/margin/repay：响应出参增加字段rawAsset，表示原始币种。

2022-06-20

接口 GET /api/v3/ticker 变动

- 权重从每symbol 5 降低到 2.
- 每次请求最多可以有100个交易对。
 - 如果symbols请求超过100个交易对, 会收到如下错误信息:

```
{
  "code": -1101,
  "msg": "Too many values sent for parameter 'symbols', maximum allowed up to 100."
}
```

- 单请求的权重上限为100。
 - 比如, 如果请求的交易对超过50个, 请求的权重是100.

2022-06-15

注意: 此变动不会立刻可用, 会在后面几天上线。

SPOT API

- 添加新接口 GET /api/v3/ticker
 - 基于windowSize 返回最近的价格变动。
 - 无需像 GET /api/v3/ticker/24hr 提供symbols参数。
 - 如果不提供 windowSize 参数, 默认值是1d。
 - 响应和 GET /api/v3/ticker/24hr 相似, 但不包括以下数据: prevClosePrice, lastQty, bidPrice, bidQty, askPrice, askQty
- 添加新接口 POST /api/v3/order/cancelReplace
 - 撤消当前的挂单并在同样的交易对上下新订单。
 - 过滤器会在撤单前做判断。
 - 例如, MAX_NUM_ORDERS 是 10, 如果目前挂单也是10, 调用 POST /api/v3/order/cancelReplace 会失败。撤单与下单的操作都不会被执行。
 - 更新将在几天后上线, 升级完毕后才会开启此功能。
- GET /api/v3/exchangeInfo 在symbols列表里返回新数据cancelReplaceAllowed。
- 添加新的过滤器 NOTIONAL
 - 基于minNotional 与 maxNotional 值来限制名义价值 (price * quantity)
- 添加新的过滤器 EXCHANGE_MAX_NUM_ICEBERG_ORDERS
 - 账号最大冰山挂单数

WEBSOCKETS

- 新的symbol ticker流, 可以选择1h 或者 4h 时间窗口 :
 - 单个交易对: <symbol>@ticker_<window-size>
 - 市场所有交易对: !ticker_<window-size>@arr

2022-06-02

- 更新子母账户接口:
 - GET /sapi/v1/sub-account/sub/transfer/history : fromEmail及toEmail可以是母账户email。

2022-05-27

- 更新法币接口 :
 - GET /sapi/v1/fiat/orders : 权重自 UID(3000) 改至 UID(90000)
- 更新Pay接口 :
 - GET /sapi/v1/pay/transactions : 参数名称改变: startTimestamp -> startTime ; endTimestamp -> endTime

2022-05-26

- 更新法币接口 :
 - GET /sapi/v1/fiat/orders : 权重自 IP(1) 改至 UID(3000)
- 更新杠杆账户接口 : 查询时间范围最大不得超过30天 :
 - GET /sapi/v1/margin/transfer
 - GET /sapi/v1/margin/loan
 - GET /sapi/v1/margin/repay
 - GET /sapi/v1/margin/isolated/transfer
 - GET /sapi/v1/margin/interestHistory

2022-05-23

- Order Book 深度的变动
 - 之前深度的数量在一些极端情况下会出现负数.
 - 之后深度数量不会溢出, 而是限制在64位的最大值, 这表示深度的数量达到 , 或者超过了最大值. 最大值和交易对的base asset的精度有关. 比如如果精度是8位小数 , 最大值则为92,233,720,368.54775807.
 - 原有的深度价位, 在修复上线后, 需要价位上有变动, 才能体现新的修复.
- 哪里有影响?
 - 现货深度接口
 - GET /api/v3/depth
 - Websocket Streams
 - <symbol>@depth
 - <symbol>@depth@100ms
 - <symbol>@depth<levels>
 - <symbol>@depth<levels>@100ms
- MAX_POSITION 的更新
 - 如果一个订单的数量(quantity) 可能导致持有仓位溢出, 会触发过滤器 MAX_POSITION.

2022-05-19

- 更新矿池接口参数:
 - GET /sapi/v1/mining/pub/algoList 及 GET /sapi/v1/mining/pub/coinList : 不需要参数
- 新增统一帐户相关错误代码 (21xxx) : -21001, -21002, -21003

2022-05-17

- GET api/v3/aggTrades 更新

- 如果同时提供 `startTime` 和 `endTime`, 旧的记录会返回.
- 如果接口 `GET /api/v3/myTrades` 中没有提供参数 `symbol`, 错误消息变为:

```
{
  "code": -1102,
  "msg": "Mandatory parameter 'symbol' was not sent, was empty/null, or malformed."
}
```

- 下面的接口提供参数 `symbols` 用于查询多个symbol.
 - `GET /api/v3/ticker/24hr`
 - `GET /api/v3/ticker/price`
 - `GET /api/v3/ticker/bookTicker`
- 上面接口的权重取决于请求 `symbols` 的数量, 具体请看下面的列表:

接口	Symbols的数量	权重
<code>GET /api/v3/ticker/price</code>	Any	2
<code>GET /api/v3/ticker/bookTicker</code>	Any	2
<code>GET /api/v3/ticker/24hr</code>	1-20	1
<code>GET /api/v3/ticker/24hr</code>	21-100	20
<code>GET /api/v3/ticker/24hr</code>	>= 101	40

2022-05-05

- 新增Binance Code接口:
 - `GET /sapi/v1/giftcard/cryptography/rsa-public-key`, 以查询RSA public key。
- 更新Binance Code接口:
 - `POST /sapi/v1/giftcard/redeemCode`: 新增参数 `externalUid`。每个外部用户 ID 代表合作伙伴平台上的某个用户。该功能帮助您识别不同用户的兑现行为。

2022-04-28

- 新增Staking接口:
 - `GET /sapi/v1/staking/productList` 以查询Staking可锁仓产品列表
 - `POST /sapi/v1/staking/purchase` 以锁仓Staking产品
 - `POST /sapi/v1/staking/redeem` 以赎回Staking产品
 - `GET /sapi/v1/staking/position` 以查询Staking产品的持仓
 - `GET /sapi/v1/staking/stakingRecord` 以查询锁仓产品的历史记录
 - `POST /sapi/v1/staking/setAutoStaking` 以设置Staking产品的自动续期
 - `GET /sapi/v1/staking/personalLeftQuota` 以查询个人锁仓限额

2022-04-27

- 新增合约策略交易接口 :
 - `POST /sapi/v1/algo/futures/newOrderTwap` 以支持合约Twap策略下单

FAQ: 时间加权平均价格策略(Twap) 介绍

2022-04-26

- 新增接口 `GET /sapi/v1/margin/rateLimit/order`
 - 回传用户在当前时间区间内的杠杆账户下单总数

2022-04-20

- 新增统一账户接口:
 - `GET /sapi/v1/portfolio/account` 以支持查询统一账户信息

FAQ: 币安合约统一账户总览

目前仅对特定用户开放此功能，详情：加入统一账户计划

2022-04-19

- 更新币安宝接口:
 - 新增返回参数 `avgAnnualInterestRate` 和 `tierAnnualInterestRate` 于接口 `GET /sapi/v1/lending/daily/product/list` 和 `GET /sapi/v1/lending/daily/token/position` 以支持查询阶梯利率

2022-04-13

- 新增合约策略交易接口：
 - `POST /sapi/v1/algo/futures/newOrderVp` 以支持合约vp策略下单
 - `DELETE /sapi/v1/algo/futures/order` 以支持合约策略委托撤单
 - `GET /sapi/v1/algo/futures/openOrders` 以支持查询合约策略当前委托
 - `GET /sapi/v1/algo/futures/historicalOrders` 以支持查询合约策略历史订单
 - `GET /sapi/v1/algo/futures/subOrders` 以支持查询合约策略子订单

FAQ: 成交量份额参与算法(VP) 介绍

2022-04-13

支持追踪止损订单

REST API

- 现货交易支持追踪止损(Trailing Stop)订单.
 - 追踪止损通过一个新的参数 `trailingDelta` 来设置基于市场价的一个自动触发价格.
 - 只适用于订单类型: `STOP_LOSS`, `STOP_LOSS_LIMIT`, `TAKE_PROFIT`, `TAKE_PROFIT_LIMIT`.
 - 参数 `trailingDelta` 的单位为基点(BIPS).
 - 比如一个 `STOP_LOSS` 卖单设置 `trailingDelta` 为 100, 那么订单会在当前市场价格从下单后的最高点下降1%的时候被触发. ($100 / 10,000 \Rightarrow 0.01 \Rightarrow 1\%$)
 - 用于OCO订单的时候, 如果市场变动触发了 `STOP_LOSS` 订单, 那么此止损订单变成追踪止损订单.
 - 当参数 `trailingDelta` 和 `stopPrice` 一起使用时, 一旦 `stopPrice` 条件被触发, 系统会开始追踪当前的价格变动. 从 `stopPrice` 价格开始, 到基于 `trailingDelta` 值之间变动.
 - 如果没有提供 `stopPrice`, 系统开始追踪价格从最新价到基于 `trailingDelta` 值之间变动.
- `POST /api/v3/order` 变动
 - 添加新可选参数 `trailingDelta`
- `POST /api/v3/order/test` 变动
 - 添加新可选参数 `trailingDelta`
- `POST /api/v3/order/oco` 变动
 - 添加新可选参数 `trailingDelta`
- 添加新的过滤器 `TRAILING_DELTA`
 - 用于限定 `trailingDelta` 的最大和最小值.

USER DATA STREAM

- User Data Stream 的 `executionReport` 添加新参数
 - "d" 代表 `trailingDelta`

2022-04-12

Note: 下面的变更会在后面几天上线.

- `GET api/v3/allOrders` 如果没有提供 `symbol`, 则返回错误信息: `{ "code": -1102, "msg": "Mandatory parameter 'symbol' was not sent, was empty/null, or malformed." }`
- 修复一个错误信息中的拼写错误。如果账号被禁用了相应的权限(比如提款，交易等), 则服务器返回错误: `"This action is disabled on this account."`
- 在市场数据(market data)审计中, 发现了一些现货的聚合交易数据(aggTrades)中的问题.
 - 丢失的记录已经被补回.
 - 重复的记录被标记成无效, 具体的值设置成如下:
 - `p = '0' // price`
 - `q = '0' // qty`
 - `f = -1 // first_trade_id`
 - `l = -1 // last_trade_id`

2022-04-08

- 更新杠杆代币WEBSOCKET :
 - 更换base url为 `wss://nbstream.binance.com/lvt-p` 对于杠杆代币数据流 `<tokenName>@tokenNav` 和 `<tokenName>@nav_kline_<interval>`. 详情见: [Websocket 杠杆代币信息更新](#) and [Websocket 杠杆代币净值K线更新](#)

2022-3-29

以下更新于**3月 31, 2022 08:00 AM UTC**生效

- 更新子母账户接口 :
 - `GET /sapi/v1/sub-account/universalTransfer`

接口查询时间窗口缩短为30天；若`startTime`和`endTime`没传，则默认返回最近30天数据。

2022-03-25

- 更新子母账户接口:
 - 新增接口 `GET /sapi/v1/managed-subaccount/accountSnapshot` 以支持投资人母账户查询托管子账户资产快照

2022-03-08

- 更新子母账户接口:
 - 新增划转类型 `MARGIN`, `ISOLATED_MARGIN` 以及传参 `symbol` 于子母账户万能划转接口 `POST /sapi/v1/sub-account/universalTransfer` 以支持母账户现货账户划转到子账户杠杆全仓账户和杠杆逐仓账户

2022-02-28

- 在接口 `GET /api/v3/exchangeInfo` 中添加新字段 `allowTrailingStop`.

2022-02-22**

现货API

- 现货规则 `PRICE_FILTER` 里面的 `(price-minPrice) % tickSize == 0` 改成 `price % tickSize == 0`
- 新添加了一个规则 `PERCENT_PRICE_BY_SIDE`.
- 接口 `GET api/v3/depth` 的变动:
 - `limit` 原先必须是固定值(比如 5, 10, 20, 50, 100, 500, 1000, 5000), 现在可以是在1-5000之间的任意的正整数, 服务器会返回指定的limit数量。(比如如果设置`limit=3`, 会返回前3个最好的卖价和买价)
 - 如果 `limit` 超过5000, 服务器也最多返回5000条记录.

- 相应的, 此接口的权重变成:

Limit	Request Weight
1-100	1
101-500	5
501-1000	10
1001-5000	50

- GET `api/v3/aggTrades` 接口的变动:
 - 当同时提供参数 `startTime` 和 `endTime`, 最旧的订单会优先返回.

2022-2-18

- 更新子母账户接口:
 - 新增响应参数 `isManagedSubAccount` 和 `isAssetManagementSubAccount` 于接口 `GET /sapi/v1/sub-account/list` 以支持查询子账户是否是托管子账户或资产管理子账户

font size=4>2022-2-17

以下更新于**2月 24, 2022 08:00 AM UTC**生效

- 更新钱包接口 :
 - `GET /sapi/v1/accountSnapshot`

接口查询范围缩短为仅支持查询最近一个月数据 , 即`startTime`不支持选定最近1个月之外的时间。

2022-2-09

- 新增钱包接口:
 - `POST /sapi/v1/asset/dust-btc` 以获取可以转换成BNB的小额资产

2022-1-25

- 自**1月 28, 2022 4:00 AM UTC**起 , 您需要使用开通允许现货和杠杆交易权限的API Key调用以下接口:
 - `POST /sapi/v1/asset/dust` 小额资产转换
 - `POST /sapi/v1/lending/daily/purchase` 申购币安宝活期产品
 - `POST /sapi/v1/lending/daily/redeem` 赎回币安宝活期产品
 - `POST /sapi/v1/lending/customizedFixed/purchase` 申购币安宝定期/活动产品
 - `POST /sapi/v1/lending/positionChanged` 币安宝定期/活动持仓转活期持仓
 - `POST /sapi/v1/bswap/liquidityAdd` 币安挖矿添加流动性
 - `POST /sapi/v1/bswap/liquidityRemove` 币安挖矿移除流动性
 - `POST /sapi/v1/bswap/swap` 币安挖矿交易
 - `POST /sapi/v1/bswap/claimRewards` 币安挖矿领取奖励

2022-1-21

- 新增币安码接口:
 - `POST /sapi/v1/giftcard/createCode` 以支持创建币安码
 - `POST /sapi/v1/giftcard/redeemCode` 以支持兑现币安码
 - `GET /sapi/v1/giftcard/verify` 以支持验证币安码

2022-1-4

- 新增矿池接口:

- GET /sapi/v1/mining/payment/uid 以获取矿池账户收益列表

- 新增币安挖矿接口:

- GET /sapi/v1/bswap/unclaimedRewards 以查询未领取的奖励数量
- POST /sapi/v1/bswap/claimRewards 以领取奖励
- GET /sapi/v1/bswap/claimedHistory 以获取已领取奖励记录

2021-12-30

- 更新杠杆接口 :

- 获取杠杆利率历史接口 GET /sapi/v1/margin/interestRateHistory 移除参数 limit , 查询时间间隔更改为最大1个月

- 更新钱包接口 :

- 由于矿池钱包合并于资金账户钱包,用户万向划转接口 POST /sapi/v1/asset/transfer 的以下划转类型 MAIN_MINING, MINING_MAIN, MINING_UMFUTURE, MARGIN_MINING 和 MINING_MARGIN 将于 **1月 05, 2022 08:00 AM UTC** 停止使用

2021-12-29

- 移除交易对类型枚举
- 新增权限枚举

2021-12-24

- 更新子母账户接口:

- 新增传参 clientTranId 于子母账户万能划转接口 POST /sapi/v1/sub-account/universalTransfer 和查询子母账户万能划转历史接口 GET /sapi/v1/sub-account/universalTransfer 以支持用户自定义划转id

2021-12-03

- 新增杠杆接口:

- 新增接口 GET /sapi/v1/margin/crossMarginData 以获取全仓杠杆利率及限额
- 新增接口 GET /sapi/v1/margin/isolatedMarginData 以获取逐仓杠杆利率及限额
- 新增接口 GET /sapi/v1/margin/isolatedMarginTier 以获取逐仓档位信息

- 新增NFT接口:

- 新增接口 GET /sapi/v1/nft/history/transactions 以支持用户查询NFT资金流水历史记录
- 新增接口 GET /sapi/v1/nft/history/deposit 以支持用户查询NFT充值历史记录
- 新增接口 GET /sapi/v1/nft/history/withdraw 以支持用户查询NFT提现历史记录
- 新增接口 GET /sapi/v1/nft/user/getAsset 以支持用户查询NFT资产

2021-11-30

- 新增闪兑接口:

- 新增接口 GET /sapi/v1/convert/tradeFlow 以支持用户查询闪兑交易历史记录

- 更新返佣接口:

- 新增接口 GET /sapi/v1/rebate/taxQuery 以支持用户查询现货返佣历史记录

2021-11-19

- 新增Pay接口:
 - 新增接口 `GET /sapi/v1/pay/transactions` 以支持用户查询Pay交易历史记录
 - 更新钱包接口:
 - 新增响应参数 `info` 于接口 `GET /sapi/v1/capital/withdraw/history` 以显示提币失败原因
-

2021-11-18

以下更新于**11月 25, 2021 08:00 AM UTC**生效

- 更新钱包接口 :
 - `GET /sapi/v1/accountSnapshot`

接口查询范围缩短为仅支持查询最近半年内的数据，即startTme不支持选定最近6个月之外的时间。若您没有传入startTme和endTme，则默认返回最近7天的数据

2021-11-17

- 以下接口将于**11月 17, 2021 13:00 PM UTC**停止使用:
 - `POST /sapi/v1/account/apiRestrictions/ipRestriction` 以支持用户为API Key开启或关闭IP白名单
 - `POST /sapi/v1/account/apiRestrictions/ipRestriction/ipList` 以支持用户为API Key添加IP白名单地址列表
 - `GET /sapi/v1/account/apiRestrictions/ipRestriction` 以支持用户为API Key查询IP白名单
 - `DELETE /sapi/v1/account/apiRestrictions/ipRestriction/ipList` 以支持用户为API Key删除IP白名单地址列表
-

2021-11-16

- 新增子母账户接口:
 - `POST /sapi/v1/sub-account/subAccountApi/ipRestriction` 以支持母账户为子账户API Key开启或关闭IP白名单
 - `POST /sapi/v1/sub-account/subAccountApi/ipRestriction/ipList` 以支持母账户为子账户API Key添加IP白名单地址列表
 - `GET /sapi/v1/sub-account/subAccountApi/ipRestriction` 以支持母账户为子账户API Key查询IP白名单
 - `DELETE /sapi/v1/sub-account/subAccountApi/ipRestriction/ipList` 以支持母账户为子账户API Key删除IP白名单地址列表
-

2021-11-09

- 新增钱包接口:
 - `POST /sapi/v1/account/apiRestrictions/ipRestriction` 以支持用户为API Key开启或关闭IP白名单
 - `POST /sapi/v1/account/apiRestrictions/ipRestriction/ipList` 以支持用户为API Key添加IP白名单地址列表
 - `GET /sapi/v1/account/apiRestrictions/ipRestriction` 以支持用户为API Key查询IP白名单
 - `DELETE /sapi/v1/account/apiRestrictions/ipRestriction/ipList` 以支持用户为API Key删除IP白名单地址列表
-

2021-11-08

- 新增质押借币接口:
 - 新增查询质押借币资金流水接口 `GET /sapi/v1/loan/income` 以支持用户查询质押借币资金流水历史
-

2021-11-05

- 更新钱包接口:
 - 新增参数 `walletType` 于提币接口 `POST /sapi/v1/capital/withdraw/apply` 以支持用户选择从 `现货钱包` 或 `资金钱包` 进行提币

2021-11-04

以下更新于**11月 11, 2021 08:00 AM UTC**生效

- 更新接口：

- GET /sapi/v1/asset/transfer
- GET /sapi/v1/futures/transfer

接口查询范围缩短为仅支持查询最近半年内的数据，即startTme不支持选定最近6个月之外的时间。若您没有传入startTme和endTme，则默认返回最近7天的数据

2021-11-01

- 新增接口 GET /api/v3/rateLimit/order
 - 回传用户在当前时间区间内的下单总数
 - 此接口的权重为 20

2021-10-22

- 钱包接口更新：
 - 新增划转类型 MAIN_FUNDING, FUNDING_MAIN, FUNDING_UMFUTURE, UMFUTURE_FUNDING, MARGIN_FUNDING, FUNDING_MARGIN, FUNDING_CMFUTURE 和 CMFUTURE_FUNDING 于用户万向划转接口 POST /sapi/v1/asset/transfer 和 GET /sapi/v1/asset/transfer 以支持资金账户和现货账户，杠杆全仓账户，U本位合约账户，币本位合约账户之间相互划转
 - 由于C2C账户，币安支付、币安卡等业务合并至资金账户，用户万向划转接口 POST /sapi/v1/asset/transfer 和 GET /sapi/v1/asset/transfer 的以下划转类型 MAIN_C2C, C2C_MAIN, C2C_UMFUTURE, C2C_MINING, UMFUTURE_C2C, MINING_C2C, MARGIN_C2C, C2C_MARGIN, MAIN_PAY 和 PAY_MAIN 将于**11月 04, 2021 08:00 AM UTC** 停止使用

2021-10-14

- 以下杠杆账户接口更新返回数据的时间范围，`startTime`与`endTime`时间跨度不能超过30天，如果不传时间参数默认返回最近7天数据，如果`archived`参数为`true`，则默认返回6个月以前的最后7天数据：
 - GET /sapi/v1/margin/transfer
 - GET /sapi/v1/margin/loan
 - GET /sapi/v1/margin/repay
 - GET /sapi/v1/margin/isolated/transfer
 - GET /sapi/v1/margin/interestHistory

2021-09-18

- 新增币安挖矿接口：
 - 新增接口 GET /sapi/v1/bswap/poolConfigure 以支持查询币对池的配置信息
 - 新增接口 GET /sapi/v1/bswap/addLiquidityPreview 以支持查询添加流动性的试算
 - 新增接口 GET /sapi/v1/bswap/removeLiquidityPreview 以查询移除流动性的试算

2021-09-17

- 访问限制介绍中新增 /api/* 和 /sapi/* 相关接口限频说明

2021-09-08

- 新增以下杠杆账户接口支持杠杆逐仓账户启用限制：
 - 新增接口 DELETE /sapi/v1/margin/isolated/account 以支持杠杆逐仓账户停用
 - 新增接口 POST /sapi/v1/margin/isolated/account 以支持杠杆逐仓账户启用

- 新增接口 `GET /sapi/v1/margin/isolated/accountLimit` 以查询杠杆逐仓账户上限
 - 查询杠杆逐仓账户信息接口 `GET /sapi/v1/margin/isolated/account` 响应加入字段 "enabled" 判断账户是否启用
-

2021-09-03

- 更新钱包接口:
 - 新增响应内容 `sameAddress`, `depositDust` 和 `specialWithdrawTips` 于 `GET /sapi/v1/capital/config/getall`
`sameAddress` 表示需要输入 memo 的币种 `depositDust` 表示最小可上帐金额 `specialWithdrawTips` 表示提现时的特殊说明
 - 新增响应内容 `confirmNo` 于 `GET /sapi/v1/capital/withdraw/history` 以支持查询提现确认数
-

2021-08-27

- 更新钱包接口:
 - 新增参数 `withdrawOrderId` 于 `GET /sapi/v1/capital/withdraw/history` 以支持查询指定 `withdrawOrderId` 的提币历史记录
 - 新增响应内容 `unlockConfirm` 于 `GET /sapi/v1/capital/deposit/hisrec` 以支持查询解锁需要的网络确认次数
-

2021-08-23

- 新增杠杆账户 OCO 接口:
 - `POST /sapi/v1/margin/order/oco`
 - `DELETE /sapi/v1/margin/orderList`
 - `GET /sapi/v1/margin/orderList`
 - `GET /sapi/v1/margin/allOrderList`
 - `GET /sapi/v1/margin/openOrderList`

用法与现货账户 OCO 相同

2021-08-20

- 更新钱包接口:
 - 新增参数 `fromSymbol`, `toSymbol` 和新增划转类型 `ISOLATEDMARGIN_MARGIN`, `MARGIN_ISOLATEDMARGIN`, `ISOLATEDMARGIN_ISOLATEDMARGIN` 于接口 `POST /sapi/v1/asset/transfer` 和 `GET /sapi/v1/asset/transfer` 以支持杠杆逐仓钱包与杠杆全仓钱包之前相互划转
-

2021-08-12

- GET `api/v3/myTrades` 添加新的参数 `orderId`
-

2021-08-05

- 新增C2C接口:
 - `GET /sapi/v1/c2c/orderMatch/listUserOrderHistory` 以查询用户C2C交易历史记录
-

2021-08-05

- 币安宝接口更新:
 - `GET /sapi/v1/lending/union/purchaseRecord`
 - `GET /sapi/v1/lending/union/redemptionRecord`
 - `GET /sapi/v1/lending/union/interestHistory`

以上接口查询范围更改为：仅支持 `startTime` 和 `endTime` 查询最大间隔为30天，若 `startTime` 和 `endTime` 均未发送，则默认返回最近30天记录

2021-07-29

- 子母账户接口更新:
 - `GET /sapi/v1/sub-account/transfer/subUserHistory` 如果`startTime`和`endTime`均未发送，默认只返回最近30天数据

2021-07-27

- 新增法币接口:
 - `GET /sapi/v1/fiat/orders` 以查询用户法币充值和提币历史记录
 - `GET /sapi/v1/fiat/payments` 以查询用户法币支付（买卖）历史记录

2021-07-16

- 新增钱包接口:
 - `GET /sapi/v1/account/apiRestrictions` 以查询用户API Key权限

2021-07-09

- 新增钱包接口:
 - `POST /sapi/v1/asset/get-funding-asset` 以查询资金账户资产，目前支持查询的业务为：Binance Pay, Binance Card, Binance Gift Card, Stock Token

2021-06-24

- 钱包接口更新:
 - `GET /sapi/v1/capital/withdraw/history` 现有的`limit`参数增加默认值1000，最大值1000的限制
 - `GET /sapi/v1/capital/deposit/hisrec` 现有的`limit`参数增加默认值1000，最大值1000的限制

2021-06-17

- 币安宝接口更新:
 - `GET /sapi/v1/lending/daily/product/list` 增加新参数`current` 和`size`

2021-06-15

- 新增子母账户接口:
 - `POST /sapi/v1/managed-subaccount/deposit` 以支持投资人账户为托管子账户充值资产（仅投资人账户方使用）
 - `GET /sapi/v1/managed-subaccount/asset` 以支持投资人账户查询托管子账户资产（仅投资人账户方使用）
 - `POST /sapi/v1/managed-subaccount/withdraw` 以支持投资人账户为托管子账户提币资产（仅投资人账户方使用）

2021-06-04

从 **八月 01, 2021 02:00 AM UTC** 开始，以下WAPI接口将停止使用：

- `GET /wapi/v3/systemStatus.html`
- `POST /wapi/v3/withdraw.html`
- `GET /wapi/v3/depositHistory.html`
- `GET /wapi/v3/withdrawHistory.html`
- `GET /wapi/v3/depositAddress.html`
- `GET /wapi/v3/accountStatus.html`
- `GET /wapi/v3/apiTradingStatus.html`
- `GET /wapi/v3/userAssetDribbletLog.html`
- `GET /wapi/v3/assetDetail.html`

- GET /wapi/v3/tradeFee.html
- GET /wapi/v3/sub-account/list.html
- GET /wapi/v3/sub-account/transfer/history.html
- POST /wapi/v3/sub-account/transfer.html
- GET /wapi/v3/sub-account/assets.html

目前WAPI已从API文档中移除，为了保证您的所有交易策略顺利执行，强烈建议所有API用户尽快更新交易程序，替换成现有的SAPI接口

2021-05-26

- 更新钱包接口：
 - 用户双向划转接口 POST /sapi/v1/asset/transfer 和 GET /sapi/v1/asset/transfer 新增划转类型 MAIN_PAY , PAY_MAIN 以支持现货和支付账户之间相互划转

2021-05-12

- 在文档中添加接口的数据来源说明
- 在每个接口中添加相应的数据源
- GET /api/v3/exchangeInfo 现在支持单或多交易对查询

2021-04-28

从 May 15, 2021 08:00 UTC 开始, 以下创建逐仓杠杆账户接口将关闭:

- POST /sapi/v1/margin/isolated/create

后续, 用户可通过逐仓杠杆账户划转 POST /sapi/v1/margin/isolated/transfer 直接完成逐仓杠杆账户的创建与交易准备, 无需调用接口创建账户

2021-04-26

从 April 28, 2021 00:00 UTC 开始, 下面接口的权重有如下变动:

- GET /api/v3/order 权重改为 2
- GET /api/v3/openOrders 权重改为 3
- GET /api/v3/allOrders 权重改为 10
- GET /api/v3/orderList 权重改为 2
- GET /api/v3/openOrderList 权重改为 3
- GET /api/v3/account 权重改为 10
- GET /api/v3/myTrades 权重改为 10
- GET /api/v3/exchangeInfo 权重改为 10

2021-04-08

- 子母账户接口更新:
 - GET /sapi/v1/sub-account/futures/accountSummary 和 GET /sapi/v2/sub-account/futures/accountSummary 接口返回字段 asset 更新为以USD计价的资产汇总, 即子账户USDT, BUSD等保证金总和

2021-04-02

- 新增钱包接口:
 - GET /sapi/v1/system/status 以获取系统状态
 - GET /sapi/v1/account/status 以获取账户状态
 - GET /sapi/v1/account/apiTradingStatus 以获取账户API交易状态
 - GET /sapi/v1/asset/dribblet 以获取小额资产转换BNB历史
 - GET /sapi/v1/asset/assetDetail 以获取上架资产详情

- 新增子母账户接口:
 - `GET /sapi/v1/asset/tradeFee` 以获取交易手续费率查询

2021-04-01

- 子母账户接口更新:
 - `GET /sapi/v1/sub-account/transfer/subUserHistory` 新增返回字段 `fromAccountType` 和 `toAccountType` 为用户转出账户类型和转入账户类型

2021-03-31

- 子母账户接口更新:
 - `GET /wapi/v3/sub-account/transfer/history.html` 新增参数 `fromEmail` 和 `toEmail`，原有参数 `email` 将默认查询 `fromEmail` 的记录

2021-03-08

- 新增子母账户接口 :
 - `POST /sapi/v1/sub-account/virtualSubAccount` 以支持母账户创建虚拟子账户
 - `GET /sapi/v1/sub-account/list` 以支持查询子账户列表
 - `POST /sapi/v1/sub-account/blvt/enable` 以支持为子账户开通杠杆代币

2021-03-05

- 新增杠杆接口 :
 - `GET /sapi/v1/margin/interestRateHistory` 以支持杠杆利率历史查询

2021-02-08

- 新增合约接口 :
 - `GET /sapi/v2/futures/loan/wallet` 混合保证金钱包 V2 接口，以支持 BUSD 借款查询
 - `GET /sapi/v2/futures/loan/configs` 混合保证金信息 V2 接口，以支持 BUSD 借款查询
 - `GET /sapi/v2/futures/loan/calcAdjustLevel` 计算调整后的混合保证金质押率 V2 接口，以支持 BUSD 借款查询
 - `GET /sapi/v2/futures/loan/calcMaxAdjustAmount` 可供调整混合保证金质押率的最大额 V2 接口，以支持 BUSD 借款质押率的调整
 - `POST /sapi/v2/futures/loan/adjustCollateral` 调整混合保证金质押率 V2 接口，以支持 BUSD 借款质押率的调整
- 更新合约接口 :
 - `GET /sapi/v1/futures/loan/adjustCollateral/history` 混合保证金调整质押率历史接口，加入参数与响应字段 `loanCoin` 以支持 BUSD 借款查询
 - `GET /sapi/v1/futures/loan/liquidationHistory` 混合保证金强平历史历史接口，加入参数与响应字段 `loanCoin` 以支持 BUSD 借款查询

2021-02-04

- 更新钱包接口 :
 - 用户万向划转接口 `POST /sapi/v1/asset/transfer` 和 `GET /sapi/v1/asset/transfer` 新增划转类型 `MARGIN_MINTING`, `MINING_MARGIN`, `MARGIN_C2C`, `C2C_MARGIN`, `MARGIN_CMFUTURE`, `CMFUTURE_MARGIN` 以支持全仓杠杆，矿池，C2C，币本位合约账户间划转。

2021-01-15

- 杠杆交易添加新接口 `DELETE /sapi/v1/margin/openOrders`
 - 此接口便于用户撤销单一交易对的所有挂单，包括OCO的挂单。

2021-01-10

- 矿池接口 `GET /sapi/v1/mining/payment/list` 新增可选参数 `pageSize`
- 矿池接口 `GET /sapi/v1/mining/payment/list` 新增返回字段：
 - "type" 表示收益类型
 - "hashTransfer" 表示已转让算力
 - "transferAmount" 表示已转让收益
- 新增矿池接口：
 - `GET /sapi/v1/mining/payment/other`
 - `GET /sapi/v1/mining/hash-transfer/config/details`
 - `GET /sapi/v1/mining/hash-transfer/config/details/list`
 - `GET /sapi/v1/mining/hash-transfer/profit/details`
 - `POST /sapi/v1/mining/hash-transfer/config`
 - `POST /sapi/v1/mining/hash-transfer/config/cancel`

2021-01-01

USER DATA STREAM

- 移除 `outboundAccountInfo` 事件。

2020-12-30

- 新增钱包接口：
 - `POST /sapi/v1/asset/transfer` 用户万向划转接口，以支持现货，全仓杠杆，合约，C2C，矿池账户间划转。
 - `GET /sapi/v1/asset/transfer` 以支持查询用户万向划转历史记录。

2020-12-22

- 新增子母账户接口：
 - `GET /sapi/v1/sub-account/sub/transfer/history` 以支持查询子母账户现货资金划转历史。

2020-12-11

- 更新合约混合保证金接口：
 - 接口 `GET /sapi/v1/futures/loan/wallet` 新增返回参数 `interestFreeLimit` 表示混合保证金总免息额度，`interestFreeLimitUsed` 表示占用混合保证金免息额度。
 - 接口 `GET /sapi/v1/futures/loan/interestHistory` 新增返回参数 `interestFreeLimitUsed` 表示占用混合保证金免息额度。

2020-12-04

- 更新杠杆代币接口：
 - 接口 `GET /sapi/v1/blvt/tokenInfo` 新增返回参数 `currentBaskets` (包括 `symbol`, `amount`, `notionalValue`)，`purchaseFeePct` 申购费率，`dailyPurchaseLimit` 每日申购数量上限，`redeemFeePct` 赎回费率，`dailyRedeemLimit` 每日赎回数量上限。
- 新增杠杆代币接口：
 - `GET /sapi/v1/blvt/userLimit` 以查询用户每日申购赎回限额。

2020-12-02

- 新增子母账户接口:
 - GET /sapi/v2/sub-account/futures/account 以支持查询子账户USDT合约和币本位合约账户详情。
 - GET /sapi/v2/sub-account/futures/accountSummary 以支持查询子账户USDT合约和币本位合约账户汇总。
 - GET /sapi/v2/sub-account/futures/positionRisk 以支持查询子账户USDT合约和币本位合约持仓信息。

2020-12-01

- 更新杠杆交易接口:
 - POST /sapi/v1/margin/order 加入参数 quoteOrderQty 支持"报价总额市价单"。

2020-11-27

为了优化性能，除了当前的 `api.binance.com`，新加了一些API的集群。如果访问 `api.binance.com` 有性能问题，也可以尝试访问：

- `https://api1.binance.com/api/v3/*`
- `https://api2.binance.com/api/v3/*`
- `https://api3.binance.com/api/v3/*`

2020-11-16

- 更新杠杆接口加入 `archived` 参数以支持查询6个月以前数据:
 - GET /sapi/v1/margin/loan
 - GET /sapi/v1/margin/repay
 - GET /sapi/v1/margin/interestHistory

2020-11-13

- 新增子母账户接口:
 - POST /sapi/v1/sub-account/universalTransfer 以支持子母账户，现货和合约账户之间相互划转。
 - GET /sapi/v1/sub-account/universalTransfer 以查询划转记录。

2020-11-10

- 新增BNB抵扣开关接口:
 - POST /sapi/v1/bnbBurn BNB现货交易和杠杆利息抵扣开关。
 - GET /sapi/v1/bnbBurn 获取BNB抵扣开关状态。

2020-11-09

- 新增返回字段 `tranId` 于子母账户接口:
 - GET /sapi/v1/sub-account/futures/internalTransfer
 - GET /sapi/v1/sub-account/transfer/subUserHistory

2020-11-03

- 更新合约接口:
 - 接口 GET /sapi/v1/futures/loan/repay/history 新增返回参数 `repayType` (`NORMAL`为混合保证金普通还款，`COLLATERAL`为抵押物还款)，`price` (抵押物还款兑换比率)，`repayCollateral` (还款所用抵押物数量)。
 - 接口 GET /sapi/v1/futures/loan/wallet 新增返回参数 `totalInterest` (混合保证金总利息)，`principalForInterest` (混合保证金计息本金)，`interest` (混合保证金利息)。
 - 接口 GET /sapi/v1/futures/loan/configs 新增返回参数 `interestRate` (混合保证金利率)，`interestGracePeriod` (混合保证金免息天数)。

- 新增合约接口:

- 接口 `GET /sapi/v1/futures/loan/collateralRepayLimit` 以查询混合保证金抵押物还款上下限。
 - 接口 `GET /sapi/v1/futures/loan/collateralRepay` 以获取混合保证金抵押物还款兑换比率。
 - 接口 `POST /sapi/v1/futures/loan/collateralRepay` 混合保证金以抵押物还款。
 - 接口 `GET /sapi/v1/futures/loan/collateralRepayResult` 以查询混合保证金以抵押物还款结果。
 - 接口 `GET /sapi/v1/futures/loan/interestHistory` 以查询混合保证金利息收取历史。
-

2020-10-14

- 合约接口更新:
 - 接口 `POST /sapi/v1/futures/loan/borrow` 与 `GET /sapi/v1/futures/loan/borrow/history` 返回新字段 `borrowId` 为用户混合保证金借款唯一 ID。
 - 接口 `POST /sapi/v1/futures/loan/repay` 与 `GET /sapi/v1/futures/loan/repay/history` 返回新字段 `repayId` 为用户混合保证金还款唯一 ID。
-

2020-10-10

- 子母账户接口 `POST /sapi/v1/sub-account/futures/transfer` 新增划转类型 `type` 以支持子账户现货账户和币本位合约账户间相互划转。
-

2020-09-30

- 杠杆账户接口更新:
 - 接口 `GET /sapi/v1/margin/maxBorrowable` 返回新字段 `borrowLimit` 为用户账户借贷限额。
-

2020-09-28

- 新增币安宝接口:
 - 接口 `POST /sapi/v1/lending/positionChanged` 以支持定期/活动持仓转成活期持仓。
 - 以下币安宝接口，lendingType里参数 `ACTIVITY` 替换 `REGULAR` 以代表币安宝活动产品：
 - 接口 `GET /sapi/v1/lending/project/list`
 - 接口 `POST /sapi/v1/lending/customizedFixed/purchase`
 - 接口 `GET /sapi/v1/lending/project/position/list`
 - 接口 `GET /sapi/v1/lending/union/purchaseRecord`
 - 接口 `GET /sapi/v1/lending/union/interestHistory`
-

2020-09-23

- 新增币安挖矿接口:
 - 接口 `GET /sapi/v1/bswap/pools` 以从某个资金池移除流动性。
 - 接口 `GET /sapi/v1/bswap/liquidity` 以获取流动资金池具体信息。
 - 接口 `POST /sapi/v1/bswap/liquidityAdd` 以添加流动性。
 - 接口 `POST /sapi/v1/bswap/liquidityRemove` 以移除流动性。
 - 接口 `GET /sapi/v1/bswap/liquidityOps` 以获取流动性操作记录。
 - 接口 `GET /sapi/v1/bswap/quote` 以获取报价。
 - 接口 `POST /sapi/v1/bswap/swap` 以交易。
 - 接口 `GET /sapi/v1/bswap/swap` 以获取交易记录。
-

2020-09-16

- 新增杠杆代币接口:

- 接口 `GET /sapi/v1/blvt/tokenInfo` 以查询杠杆代币信息。
 - 接口 `POST /sapi/v1/blvt/subscribe` 以申购代币。
 - 接口 `GET /sapi/v1/blvt/subscribe/record` 以查询申购代币记录。
 - 接口 `POST /sapi/v1/blvt/redeem` 以赎回代币。
 - 接口 `GET /sapi/v1/blvt/redeem/record` 以查询赎回代币记录。
- 以下杠杆代币功能请使用合约接口：
 - 杠杆代币历史净值K线接口。
 - WebSocket 杠杆代币信息更新和净值K线更新

2020-09-09

用户数据 STREAM

- `outboundAccountInfo` 事件不再推荐使用。
- `outboundAccountInfo` 事件以后会被删除(具体时间未定) 请使用 `outboundAccountPosition` 事件。
- `outboundAccountInfo` 只推送余额不为0，以及余额刚变成0的资产。

2020-09-03

- 新增子母账户接口 `POST /sapi/v1/sub-account/futures/internalTransfer` 以执行子账户合约资金直接划转。
- 新增子母账户接口 `GET /sapi/v1/sub-account/futures/internalTransfer` 以查询子账户合约资金直接划转历史。

2020-09-01

- 子母账户接口 `GET /sapi/v1/sub-account/spotSummary` 返回内容中新增字段 `masterAccountTotalAsset` 以获取BTC计价的母账户资产。

2020-08-27

- 新增接口 `GET /sapi/v1/sub-account/spotSummary` 以获取BTC计价的子账户现货资产汇总。

2020-08-26

- 逐仓杠杆接口 `GET /sapi/v1/margin/isolated/account` 新增可选参数 `symbols`, 以支持查询至多5个指定symbol的杠杆逐仓资产。

2020-07-28

逐仓杠杆相关接口

- 以下接口新增可选参数"isolated", 并在返回内容中新增字段 "symbol":
 - `POST /sapi/v1/margin/loan`
 - `POST /sapi/v1/margin/repay`
- 以下接口新增可选参数"isolated", 并在返回内容中新增字段 "isIsolated":
 - `POST /sapi/v1/margin/order`
 - `DELETE /sapi/v1/margin/order`
 - `GET /sapi/v1/margin/order`
 - `GET /sapi/v1/margin/openOrders`
 - `GET /sapi/v1/margin/allOrders`
 - `GET /sapi/v1/margin/myTrades`
- 以下接口新增可选参数"isolatedSymbol", 并在返回内容中新增字段 "isolatedSymbol":
 - `POST /sapi/v1/margin/order`
 - `DELETE /sapi/v1/margin/order`
 - `GET /sapi/v1/margin/order`
 - `GET /sapi/v1/margin/openOrders`
 - `GET /sapi/v1/margin/allOrders`
 - `GET /sapi/v1/margin/myTrades`

- GET /sapi/v1/margin/loan
- GET /sapi/v1/margin/repay
- GET /sapi/v1/margin/interestHistory

- 接口 GET /sapi/v1/margin/forceLiquidationRec 新增可选参数"isolatedSymbol", 并在返回内容中新增字段 "isIsolated"
- 以下接口新增可选参数"isolatedSymbol":

 - GET /sapi/v1/margin/maxBorrowable
 - GET /sapi/v1/margin/maxTransferable

- 新增以下逐仓杠杆功能接口:

 - POST /sapi/v1/margin/isolated/create
 - POST /sapi/v1/margin/isolated/transfer
 - GET /sapi/v1/margin/isolated/transfer
 - GET /sapi/v1/margin/isolated/account
 - GET /sapi/v1/margin/isolated/pair
 - GET /sapi/v1/margin/isolated/allPairs

- 新增以下接口 , 管理逐仓杠杆账户listenKey:

 - POST /sapi/v1/userDataStream/isolated
 - PUT /sapi/v1/userDataStream/isolated
 - DELETE /sapi/v1/userDataStream/isolated

2020-07-20

- 接口 GET /sapi/v1/margin/allOrders 参数"limit"的可传最大值更新为500.

2020-07-17

- 接口 GET /sapi/v1/margin/allOrders 增加访问限制为每个IP最多每分钟60次

2020-07-13

- 新增合约混合保证金相关的SAPI接口:

 - POST /sapi/v1/futures/loan/borrow
 - GET /sapi/v1/futures/loan/borrow/history
 - POST /sapi/v1/futures/loan/repay
 - GET /sapi/v1/futures/loan/repay/history
 - GET /sapi/v1/futures/loan/wallet
 - GET /sapi/v1/futures/loan/configs
 - GET /sapi/v1/futures/loan/calcAdjustLevel
 - GET /sapi/v1/futures/loan/calcMaxAdjustAmount
 - POST /sapi/v1/futures/loan/adjustCollateral
 - GET /sapi/v1/futures/loan/adjustCollateral/history
 - GET /sapi/v1/futures/loan/liquidationHistory

2020-06-28

- 服务于合约的相关SAPI接口内容转移至本文档:

 - POST /sapi/v1/futures/transfer
 - GET /sapi/v1/futures/transfer

2020-05-06

- 新增矿池接口:

- GET /sapi/v1/mining/pub/algoList
- GET /sapi/v1/mining/pub/coinList
- GET /sapi/v1/mining/worker/detail
- GET /sapi/v1/mining/worker/list
- GET /sapi/v1/mining/payment/list
- GET /sapi/v1/mining/statistics/user/status
- GET /sapi/v1/mining/statistics/user/list

2020-05-01

- 从2020-05-01 UTC 00:00开始, 所有交易对都会有最多200个挂单的限制, 体现在过滤器MAX_NUM_ORDERS上.
 - 已经存在的挂单不会被移除或者撤销。
 - 单交易对(symbol)的挂单数量达到或超过200的账号, 无法在此交易对上下新的订单, 除非挂单数量低于200。
 - OCO订单在被触发成LIMIT订单, 或者被触发成STOP_LOSS(或者STOP_LOSS_LIMIT)前, 被认为是2个挂单量. 一旦OCO订单被触发, 就只被算作一个挂单。

2020-04-25

现货 API

- 添加新字段 permissions
 - 这个字段定义了对于账户、交易对(symbol)的交易权限。
 - permissions 是个enum数组, 可能的值:
 - SPOT
 - MARGIN
 - 在未来的版本(v4)中, permissions 将会在 GET api/v3/exchangeInfo 中替换 isSpotTradingAllowed 和 isMarginTradingAllowed。
 - 如果账户想在一个交易对下做交易, 账户和交易对必须同时拥有对应的权限。
- 接口 GET api/v3/exchangeInfo 的更新
 - 添加新字段 permissions。
 - 添加新字段 quoteAssetPrecision。此字段和 quotePrecision 重复。在未来的版本(v4)中 quotePrecision 会被移除。
- 接口 GET api/v3/account 的更新
 - 添加新字段 permissions。
- 添加新接口 DELETE api/v3/openOrders
 - 此接口便于用户撤销单一交易对的所有挂单, 包括OCO的挂单。
- 如果交易对处于 BREAK 或者 HALT 状态, 挂单也可以被撤销。

用户数据 STREAM

- OutboundAccountInfo 消息会显示一个新字段 P, 用来显示账户的交易权限。

2020-04-23

WEB SOCKET 连接限制

- WebSocket服务器每秒最多接受5个消息。消息包括:
 - PING帧
 - PONG帧
 - JSON格式的消息, 比如订阅, 断开订阅。
- 如果用户发送的消息超过限制, 连接会被断开连接。反复被断开连接的IP有可能被服务器屏蔽。
- 单个连接最多可以订阅 1024 个Streams。

2020-04-16

- 币安宝接口 GET /sapi/v1/lending/daily/token/position 返回内容新增字段 :

- `todayPurchasedAmount` 表示用户今日申购的活期产品数量

- 新增以下币安宝接口用以支持灵活定期产品:

- `GET /sapi/v1/lending/project/list`
- `POST /sapi/v1/lending/customizedFixed/purchase`
- `GET /sapi/v1/lending/project/position/list`

2020-04-02

- 接口 `GET /sapi/v1/capital/config/getall` 返回内容新增字段 :
 - `minConfirm` 表示资产上账所需的最小确认数
 - `unLockConfirm` 表示资产解锁需所需确认数

2020-03-24

- 添加过滤器 `MAX_POSITION`.
 - 这个过滤器定义账户允许的基于 `base asset` 的最大仓位。一个用户的仓位可以定义为如下资产的总和:
 - `base asset` 的可用余额
 - `base asset` 的锁定余额
 - 所有处于open的买单的数量总和
 - 如果用户的仓位大于最大的允许仓位，买单会被拒绝。

2020-03-13

- 新增可选参数 `transactionFeeFlag` 于以下提币接口:
 - `POST /sapi/v1/capital/withdraw/apply`
 - `POST /wapi/v3/withdraw.html`

2020-02-05

- 新增子账户相关接口:
 - `POST /sapi/v1/sub-account/futures/transfer`: 对子账户实施futures账户划转
 - `POST /sapi/v1/sub-account/margin/transfer`: 对子账户实施margin账户划转
 - `POST /sapi/v1/sub-account/transfer/subToSub`: 向兄弟子账户划转
 - `POST /sapi/v1/sub-account/transfer/subToMaster`: 向母账户划转
 - `GET /sapi/v1/sub-account/transfer/subUserHistory`: 子账户获取自身划转历史

2020-01-15

- 接口 `POST /wapi/v3/withdraw.html` 新增参数 `withdrawOrderId`: 用户自定义提币id
- 接口 `GET /wapi/v3/withdrawHistory.html` 返回内容新增字段 `withdrawOrderId`: 该笔提币的用户自定义id

2019-12-25

- 新增币安宝接口 :
 - `GET /sapi/v1/lending/daily/product/list`
 - `GET /sapi/v1/lending/daily/userLeftQuota`
 - `POST /sapi/v1/lending/daily/purchase`
 - `GET /sapi/v1/lending/daily/userRedemptionQuota`
 - `POST /sapi/v1/lending/daily/redeem`
 - `GET /sapi/v1/lending/daily/token/position`
 - `GET /sapi/v1/lending/union/account`

- GET /sapi/v1/lending/union/purchaseRecord
- GET /sapi/v1/lending/union/redemptionRecord
- GET /sapi/v1/lending/union/interestHistory

- 新增请求时间间隔于以下接口

GET /sapi/v1/capital/withdraw/history,
 GET /wapi/v3/withdrawHistory.html,
 GET /sapi/v1/capital/deposit/hisrec and
 GET /wapi/v3/depositHistory.html:

- 默认 startTime 为当前时间起90天前， 默认 endTime 为当前时间；
- 请注意 startTime 与 endTime 的默认时间戳，保证请求时间间隔不超过90天；
- 同时提交 startTime 与 endTime 间隔不得超过90天.

2019-12-18

- 新增接口用以获取账户每日资产快照:

GET /sapi/v1/accountSnapshot

2019-11-30

- 接口 POST /sapi/v1/margin/order (HMAC SHA256) 新增参数 sideEffectType , 可选内容如下:
 - NO_SIDE_EFFECT: 普通交易订单;
 - MARGIN_BUY: 自动借款交易订单;
 - AUTO_REPAY: 自动还款交易订单.
- New field marginBuyBorrowAmount and marginBuyBorrowAsset in FULL response to POST /sapi/v1/margin/order (HMAC SHA256)

2019-11-28

- 新增SAPI接口用以关闭账户站内划转功能 :

POST /sapi/v1/account/disableFastWithdrawSwitch (HMAC SHA256)
- 新增SAPI接口用以开启账户站内划转功能 :

POST /sapi/v1/account/enableFastWithdrawSwitch (HMAC SHA256)

2019-11-22

- "报价总额市价单"作为新的市价单方式已在各交易对投入使用。
 - "报价总额市价单" 允许用户在市价单 MARKET 中设置总的购买投入金额或卖出预计回收金额 quoteOrderQty。
 - "报价总额市价单"不会突破 LOT_SIZE 的限制规则; 报单会按给定的 quoteOrderQty 尽可能接近地被执行。
 - 以BNBBTC交易对为例:
 - On the BUY side, the order will buy as many BNB as quoteOrderQty BTC can.
 - 买单: 给定 quoteOrderQty 的BTC会被用来市价买入尽可能多的BNB。
 - On the SELL side, the order will sell as much BNB as needed to receive quoteOrderQty BTC.
 - 卖单: 持有BNB会被尽可能多地以市价卖出以获取给定 quoteOrderQty 的BTC。

2019-11-19

- GET /sapi/v1/sub-account/margin/account 返回内容新增: marginTradeCoeffVo 其中包括
 - forceLiquidationBar: 强平风险率;
 - marginCallBar: 补仓风险率;
 - normalBar: 初始风险率

2019-11-13

Rest API

- "api/v3/exchangeInfo" 新增内容:
 - `quoteOrderQtyMarketAllowed`
 - `baseCommissionPrecision`
 - `quoteCommissionPrecision`
- MARKET orders (市价单)新增可选参数: `quoteOrderQty` 指定买入或卖出的报价数量，不可与 `quantity` (数量)同时使用。
 - 能够有效配合该参数使用 MARKET orders(市价单)的确切时间和进一步详细信息将由后续声明予以通告。
- 所有订单查询接口增加新的返回内容 : `origQuoteOrderQty` (e.g. GET api/v3/allOrders)

```
{
  "code": -1128,
  "msg": "Combination of optional parameters invalid. Recommendation: 'stopLimitTimeInForce' should also be specified when using 'stopLimitPrice'."}
```

- 错误代码更新: -1128
 - 发送 OCO 订单中有 `stopLimitPrice` 但是没有 `stopLimitTimeInForce`，将会受到错误信息:
- 错误代码更新: -1003, 明确了使用请求权重作为限制而不是请求数量。

v1 接口将被弃用:

2020年一季度末，以下接口将被移除。目前文档已经将这些接口更新为v3版本。

- GET api/v1/depth
- GET api/v1/historicalTrades
- GET api/v1/aggTrades
- GET api/v1/klines
- GET api/v1/ticker/24hr
- GET api/v1/ticker/price
- GET api/v1/exchangeInfo
- POST api/v1/userDataStream
- PUT api/v1/userDataStream
- GET api/v1/ping
- GET api/v1/time
- GET api/v1/ticker/bookTicker

以下接口将不会移植到v3版本，请使用新接口予以替换

旧的 V1 接口	新的 V3 接口
GET api/v1/ticker/allPrices	GET api/v3/ticker/price
GET api/v1/ticker/allBookTickers	GET api/v3/ticker/bookTicker

USER DATA STREAM

- 事件 `executionReport` (订单更新)更新内容:
 - 如果 C 值为空, 将返回 `null`, 而不是 `"null"`.
 - 新增返回值 Q 表示 `quoteOrderQty`.
- 新增事件类型 `balanceUpdate` (余额更新)
 - 当资金存入或从帐户中提取时，发生余额更新。

WEB SOCKET STREAM

- WSS 现在支持实时订阅和取消数据流。

2019-11-08

- 新增以下sapi接口用以管理子账户的杠杆与期货：
 - GET /sapi/v1/sub-account/status (HMAC SHA256)
 - POST /sapi/v1/sub-account/margin/enable (HMAC SHA256)
 - GET /sapi/v1/sub-account/margin/account (HMAC SHA256)
 - GET /sapi/v1/sub-account/margin/accountSummary (HMAC SHA256)
 - POST /sapi/v1/sub-account/futures/enable (HMAC SHA256)
 - GET /sapi/v1/sub-account/futures/account (HMAC SHA256)
 - GET /sapi/v1/sub-account/futures/accountSummary (HMAC SHA256)
 - GET /sapi/v1/sub-account/futures/positionRisk (HMAC SHA256)

2019-11-04

- 新增管理子账户充值功能相关的sapi接口
 - GET /sapi/v1/capital/deposit/subAddress (HMAC SHA256): 获取子账户充值地址。
 - GET /sapi/v1/capital/deposit/subHisrec (HMAC SHA256): 获取子账户充值记录。

2019-10-29

- 新增钱包提币功能相关的sapi接口
 - POST /sapi/v1/capital/withdraw/apply (HMAC SHA256): 提币。
 - Get /sapi/v1/capital/withdraw/history (HMAC SHA256): 获取提币历史(支持多网络)。

2019-10-14

- 新增钱包功能相关的sapi接口
 - GET /sapi/v1/capital/config/getall (HMAC SHA256): 获取针对用户的所有币种信息。
 - GET /sapi/v1/capital/deposit/hisrec (HMAC SHA256): 获取充值历史(支持多网络)。
 - GET /sapi/v1/capital/deposit/address (HMAC SHA256): 获取充值地址(支持多网络)。

2019-10-11

- POST /wapi/v3/withdraw.html,增加参数 network, 支持多网络提币。

2019-09-09

- 新增bookTicker行情流: <symbol>@bookTicker 与 !bookTicker.

2019-09-03

- 更新频率达到100ms的更快的 order book 信息流选项: <symbol>@depth@100ms 和 <symbol>@depth#@100ms
- WebSocket Market Streams 增加 Update Speed 更新速度

2019-08-16

- 10000 限额 的接口已被临时删除： GET api/v1/depth
- 在2017年第四季度，以下接口已被弃用并将其从API文档中删除。从此版本开始，以下接口已从API中永久删除。对于原始变更日志如有遗漏，我们深表歉意：
 - GET api/v1/order
 - GET api/v1/openOrders
 - POST api/v1/order
 - DELETE api/v1/order

- GET api/v1/allOrders
- GET api/v1/account
- GET api/v1/myTrades

- 在此存储库的文档中描述的流、接口、参数、有效负载等均被 **官方认证且得到支持**。任何其他流、接口、参数或有效负载等的使用**不受支持，自行使用的风险将由您自己承担，没有任何保证**。

2019-09-15

Rest API

- 新订单类型: OCO ("One Cancels the Other")
 - 一个 OCO 有 2 个订单: (在财务术语中也称为 legs)
 - STOP LOSS 或 STOP LOSS LIMIT leg
 - LIMIT MAKER leg
 - 价格限制:
 - SELL Orders : 限价 > 成交价>止损价
 - BUY Orders : 限价<成交价<止损价
 - 如前所述, 价格必须"横跨"交易品种的最后交易价格。例如: 如果最后价格是10:
 - 卖出OCO的限制价格必须大于10, 止损价格小于10.
 - 买入OCO的限制价格必须小于10, 止损价格大于10.
 - 数量限制:
 - 两个 legs 的数量必须相同。
 - 但是, ICEBERG 的数量不必相同。
 - 执行顺序:
 - 如果触发了 LIMIT MAKER , 则在取消止损leg之前将首先执行限价支路。
 - 如果市场价格移动到将触发"STOP LOSS"或"STOP LOSS LIMIT" , 则在执行"STOP LOSS"支路之前, 限价单支路将被取消。
 - 取消一个 OCO 订单
 - 取消任一订单的 leg 将取消整个 OCO 订单.
 - 可通过 orderListId 或 listClientOrderId 取消整个 OCO 订单。
 - OCO的新枚举:
 1. ListStatusType
 - RESPONSE - 当ListStatus响应失败的操作时使用。 (下单或取消订单)
 - EXEC_STARTED - 在下订单列表或列表状态更新时使用。
 - ALL_DONE - 当订单清单完成执行且不再有效时使用。
 2. ListOrderStatus
 - EXECUTING - 在下订单列表或列表状态更新时使用。
 - ALL_DONE - 当订单清单完成执行且不再有效时使用。
 - REJECT - 当ListStatus响应失败的操作时使用。 (下单或取消订单)
 3. ContingencyType
 - OCO - 指定订单列表的类型。
 - 新的接口:
 - POST api/v3/order/oco
 - DELETE api/v3/orderList
 - GET api/v3/orderList
- recvWindow cannot exceed 60000.
- New intervalLetter values for headers:
 - SECOND => S
 - MINUTE => M
 - HOUR => H
 - DAY => D
- 新标头 "X-MBX-USED-WEIGHT-(intervalNum)(intervalLetter)" 将为(intervalNum)(intervalLetter)速率限制器提供您当前使用的请求权重。例如, 如果设置了一分钟的请求速率权重限制器, 则响应中将获得一个 "X-MBX-USED-WEIGHT-1M" 标头。旧标头 X-MBX-USED-WEIGHT 仍将返回, 并代表一分钟请求速率权重限制的当前使用权重。

- 新标头 "X-MBX-ORDER-COUNT-(intervalNum)(intervalLetter)" 会在任何有效的订单位置上更新，并跟踪该间隔的当前订单数；拒绝/不成功的订单不保证在响应中具有 X-MBX-ORDER-COUNT-** 标头。
 - 例如："X-MBX-ORDER-COUNT-1S" 用于"每1秒钟的订单"，X-MBX-ORDER-COUNT-1D 用于"每1天的订单"
- GET api / v1 / depth 现在支持 limit 5000 和 10000；权重分别是 50 和 100。
- GET api / v1 / exchangeInfo 具有一个新参数 "ocoAllowed"。

用户数据流

- `executionReport` 事件现在包含具有 `orderListId` 的 "g"；对于非OCO订单，它将设置为 -1。
- 新事件类型 `listStatus`；`listStatus` 是在更新任何 OCO 订单时发送的。
- 新事件类型 `outboundAccountPosition`；每当帐户余额发生变化时，就会发送 `outboundAccountPosition`，并包含可能导致余额发生变化的事件(存款，提款，交易，下单或取消)更改的资产。

新的错误码

- 1131 BAD_RECV_WINDOW**
 - `recvWindow` 必须小于 60000
- 1099 未被找到，被认证或被授权** * 替换错误代码-1999

新的-2011错误内容

- OCO_BAD_ORDER_PARAMS**
 - 其中一个订单的参数不正确。
- OCO_BAD_PRICES**
 - 订单价格之间的关系不正确。
- UNSUPPORTED_ORD_OCO**
 - 此交易对不支持 OCO 订单。

2019-03-12

Rest API

- X-MBX-US-WEIGHT 标头已添加到 Rest API 响应中。
- Retry-After 标头已添加到 Rest API 418 和 429 响应中。
- 取消 Rest API 时，如果交易对的状态不是 "TRADING"，则现在可以返回 "errorCode" -1013 或 -2011。`api/v1/depth` 不再具有被忽略和为空的 []。 * `api/v3/myTrades` 现在返回 `quoteQty`：价格交易数量。

Websocket 流

- `<symbol>@depth` 和 `<symbol>@depthX` 流不再具有被忽略且为空的 "[]"。

系统改进

- 匹配引擎稳定性/可靠性改进。
- Rest API 性能改进。

2018-11-13

Rest API

- 现在可以在限制交易期间通过 Rest API 取消订单。
- 新的过滤器：`PERCENT_PRICE`，`MARKET_LOT_SIZE`，`MAX_NUM_ICEBERG_ORDERS`。
- 添加了 `RAW_REQUESTS` 速率限制。限制取决于 X 分钟内的请求数量(不考虑重量)。
- 无交易对查询的 `/api/v3/ticker/price` 权重增加到 2。
- `/api/v3/ticker/bookTicker` 对于无符号查询增加了 2 的权重。
- `DELETE /api/v3/order` 现在将返回订单最终状态的执行报告。

- `MIN_NOTIONAL` 过滤器有两个新参数：
 - `applyToMarket` (过滤器是否应用于MARKET订单)
 - `avgPriceMins` (平均价格的分钟数)。
- `intervalNum` 已添加到 `/api/v1/exchangeInfo` 限制中。 `intervalNum` 描述间隔的数量。 例如：`intervalNum 5`，带有 `interval` 分钟，表示“每5分钟”。

平均价格的计算规则解释：

1. [过去5分钟所有订单的数量*价格求和] / 过去5分钟所有订单的数量
2. 如果最近5分钟内没有交易，则以5分钟窗口外发生的第一笔交易为准。 例如，如果最后一次交易是在20分钟前，则该交易的价格为5分钟的平均值。
3. 如果代码上没有交易，则没有平均价格，因此无法下达市价单。对于在`MIN_NOTIONAL`过滤器上启用了`applyToMarket`的新交易对，除非有至少一笔交易，才能下达市价单。
4. 当前的平均价格可以在这里查看：<https://api.binance.com/api/v3/avgPrice?symbol=<symbol>> 例如：<https://api.binance.com/api/v3/avgPrice?symbol=BNBUSDT>

用户数据流

- 将“最后报价资产交易量”(作为变量“Y”)添加到执行报告中。代表 `lastPrice`
- `lastQty(L * 1)`。

2018-07-18

Rest API

- 新的过滤器：`ICEBERG_PARTS`
- `post api/v3/order` 为 `newOrderRespType` 的新默认值。 `ACK`, `RESULT` 或 `FULL`；“`MARKET`” 和 “`LIMIT`” 订单类型默认为 “`FULL`”，所有其他订单默认为 “`ACK`”。
- POST `api/v3 order` `RESULT` 和 `FULL` 响应现在具有 “`cummulativeQuoteQty`”
- `GET/api/v3/ openOrders` 的交易对权重减少到 40。
- `GET/api/v3/ ticker / 24hr`，且交易对权重未降低至 40。
- `GET/api/v1/ trades` 的最大交易量增加到 1000。
- `GET/api/v1/ historicalTrades` 的最大交易量增加到 1000。
- `GET/api/v1/ aggTrades` 的最大总交易量增加到 1000。
- `GET/api/v1/ klines` 的最大总交易量增加到 1000。
- 剩余的 API 订单查询现在返回 `updateTime`，它代表订单的最后更新时间；`time` 是订单创建时间。
- 订单查找接口现在将返回 “`cummulativeQuoteQty`”。如果 “`cummulativeQuoteQty`” 小于 0，则表示该时间该数据不可用。
- REQUESTS 速率限制类型更改为 `REQUEST_WEIGHT`。从逻辑上讲，此限制始终是请求权重，并且其先前的名称引起混乱。

用户数据流

- 在订单响应和执行报告中添加了 “`cummulativeQuoteQty`” 字段(作为变量“Z”)。表示已花费(使用“买入”订单)或已收到(使用“卖出”订单)的“报价”的累计金额。历史订单在该字段中的值将小于 0，这表明该数据目前不可用。“`cummulativeQuoteQty`”除以 “`cummulativeQty`” 将得出订单的平均价格。
- `O`(订单创建时间)添加到执行报告中

2018-01-23

- `GET/api/v1/ historicalTrades` 权重降低到 5
- `GET/api/v1/ aggTrades` 权重降至 1
- `GET/api/v1/ klines` 权重降至 1
- `GET/api/v1/ticker / 24hr`，所有交易品种的权重降低到交易交易品种的数量 / 2
- `GET/api/v3/ allOrders` 权重降低到 5
- `GET/api/v3/ myTrades` 权重降低到 5
- `GET/api/v3/ 账户权重降低到 5`
- `GET/api/v1/ 深度限制 = 500` 重量减少到 5
- `GET/api/v1/ 深度限制 = 1000` 重量减少到 10

- 1003错误消息已更新，可将用户定向到websocket

2018-01-20

- GET/api/v1/ticker / 24hr单个符号权重降至1
- GET/api/v3/ openOrders所有交易对权重下降至交易交易品种数量/ 2
- GET/api/v3/allOrders权重降低到15
- GET/api/v3/ myTrades权重降低到15
- GET/api/v3/订单权重降至1
- myTrades现在将返回自交易/清洗交易的双方

2018-01-14

- GET/api/v1/aggTrades权重更改为2
- GET/api/v1/klines权重更改为2
- GET/api/v3/订单权重更改为2
- GET/api/v3/ allOrders权重更改为20
- GET/api/v3/帐户权重更改为20
- GET/api/v3/ myTrades权重更改为20
- GET/api/v3/ historicalTrades权重更改为20

介绍

API Key 设置

- 很多接口需要API Key才可以访问。请参考这个页面来设置API Key。
- 设置API Key的同时，为了安全，建议设置IP访问白名单。
- 永远不要把你的API key/secret告诉给任何人**

如果不小心泄露了API key，请立刻删除此Key，并可以另外生产新的Key。

API Key 权限设置

- 新创建的API的默认权限是只读。
- 如果需要通过API提款，需要在UI修改权限，选中允许提现。

账户

现货账户

新注册的币安账号都会有一个现货(Spot)账号。

杠杆账户

为了开设杠杆(Margin)账户，可以参考Binance杠杆交易账户设置指南

现货测试网

用户可以使用现货的测试网来体验`SPOT`交易。现在只能通过API来交易。

更多信息请参考现货测试网。

API 代码库

PYTHON CONNECTOR

一个轻量级的Python代码库，提供让用户直接调用API的方法。支持所有现货的接口。

<https://github.com/binance/binance-connector-python>

NODE.JS CONNECTOR

一个轻量级的代码库，提供Node.js用户直接调用API的方法。支持所有现货的接口。

<https://github.com/binance/binance-connector-node>

RUBY CONNECTOR

一个轻量级的代码库，提供Ruby用户直接调用API的方法。支持所有现货的接口。

<https://github.com/binance/binance-connector-ruby>

DOTNET CONNECTOR

一个轻量级的代码库，提供C#用户直接调用API的方法。支持所有现货的接口。

<https://github.com/binance/binance-connector-dotnet>

JAVA CONNECTOR

一个轻量级的代码库，提供Java用户直接调用API的方法。支持所有现货的接口。

<https://github.com/binance/binance-connector-java>

POSTMAN COLLECTIONS

现在你可以通过`Postman collection`来快速体验、使用API接口。

如果想了解更多如何使用Postman，请访问: Binance API Postman

SWAGGER

一个基于OpenAPI规范的RESTful API接口定义的YAML文件，还有便于交互的 Swagger UI 页面。

<https://github.com/binance/binance-api-swagger>

联系我们

- 币安API电报群
 - 咨询关于API或者Websockets性能方面的问题.
 - 咨询文档中没有提及的API问题.
- 币安开发者社区
 - 咨询关于API/Websockets代码实现，或者任何API/Websockets的问题.
- 币安客服
 - 咨询关于账户，钱包，2FA等.

基本信息

API 基本信息

- 接口可能需要用户的 API Key , 如何创建API-KEY请参考[这里](#)
- 本篇列出接口的baseurl: <https://api.binance.com>
- 如果上面的baseURL访问有性能问题 , 请访问下面的API集群:
 - <https://api1.binance.com>
 - <https://api2.binance.com>
 - <https://api3.binance.com>
- 所有接口的响应都是 JSON 格式。
- 响应中如有数组 , 数组元素以时间升序排列 , 越早的数据越提前。
- 所有时间、时间戳均为UNIX时间 , 单位为毫秒。

HTTP 返回代码

- HTTP [4XX](#) 错误码用于指示错误的请求内容、行为、格式。问题在于请求者。
- HTTP [403](#) 错误码表示违反WAF限制(Web应用程序防火墙)。
- HTTP [409](#) 错误码表示重新下单(cancelReplace)的请求部分成功。(比如取消订单失败 , 但是下单成功了)
- HTTP [429](#) 错误码表示警告访问频次超限 , 即将被封IP。
- HTTP [418](#) 表示收到429后继续访问 , 于是被封了。
- HTTP [5XX](#) 错误码用于指示Binance服务侧的问题。

接口错误代码

- 使用接口 [/api/v3](#), 以及 [/sapi/v1/margin](#)时, 每个接口都有可能抛出异常;

API 与 SAPI 的错误代码返回形式如下:

```
{
  "code": -1121,
  "msg": "Invalid symbol."
}
```

- 具体的错误码及其解释在 [错误代码](#).

接口的基本信息

- [GET](#) 方法的接口, 参数必须在 [query string](#)中发送。
- [POST](#), [PUT](#), 和 [DELETE](#) 方法的接口,参数可以在内容形式为[application/x-www-form-urlencoded](#)的 [query string](#) 中发送 , 也可以在 [request body](#) 中发送。如果你喜欢 , 也可以混合这两种方式发送参数。
- 对参数的顺序不做要求。
- 但如果同一个参数名在query string和request body中都有 , query string中的会被优先采用。

访问限制

访问限制基本信息

- 以下 是 [intervalLetter](#) 作为头部值:

- SECOND => S
- MINUTE => M
- HOUR => H
- DAY => D

- 在 `/api/v3/exchangeInfo` `rateLimits` 数组中包含与交易的有关RAW_REQUESTS , REQUEST_WEIGHT和ORDERS速率限制相关的对象。这些在 `限制种类 (rateLimitType)` 下的 `枚举定义` 部分中进一步定义。
- 违反任何一个速率限制时，将返回429。

IP 访问限制

- 每个请求将包含一个 `X-MBX-US-ED-WEIGHT-(intervalNum)(intervalLetter)` 的头，其中包含当前IP所有请求的已使用权重。
- 每一个接口均有一个相应的权重(weight)，有的接口根据参数不同可能拥有不同的权重。越消耗资源的接口权重就会越大。
- 收到429时，您有责任停止发送请求，不得滥用API。
- 收到429后仍然继续违反访问限制，会被封禁IP，并收到418错误码**
- 频繁违反限制，封禁时间会逐渐延长，**从最短2分钟到最长3天**。
- `Retry-After`的头会与带有418或429的响应发送，并且会给出**以秒为单位**的等待时长(如果是429)以防止禁令，或者如果是418，直到禁令结束。
- 访问限制是基于IP的，而不是API Key**

① 建议您尽可能多地使用websocket消息获取相应数据，以减少请求带来的访问限制压力。

下单频率限制

- 每个成功的下单回报将包含一个 `X-MBX-ORDER-COUNT-(intervalNum)(intervalLetter)` 的头，其中包含当前账户已用的下单限制数量。
- 当下单数超过限制时，会收到带有429但不含 `Retry-After` 头的响应。请检查 `GET api/v3/exchangeInfo` 的下单频率限制 (`rateLimitType = ORDERS`) 并等待封禁时间结束。
- 被拒绝或不成功的下单并不保证回报中包含以上头内容。
- 下单频率限制是基于每个账户计数的。**
- 用户可以通过接口 `GET api/v3/rateLimit/order` 来查询当前的下单量。

WEB SOCKET 连接限制

- Websocket服务器每秒最多接受5个消息。消息包括:
 - PING帧
 - PONG帧
 - JSON格式的消息，比如订阅, 断开订阅。
- 如果用户发送的消息超过限制，连接会被断开连接。反复被断开连接的IP有可能被服务器屏蔽。
- 单个连接最多可以订阅 **1024** 个Streams。

/API/ 与 /SAPI/ 接口限频说明

`/api/*` 接口和 `/sapi/*` 接口采用两套不同的访问限频规则, 两者互相独立。

- `/api/*` 的接口相关：
 - 按IP和按UID(account)两种模式分别统计, 两者互相独立。
 - 以 `/api/*` 开头的接口按IP限频，**且所有接口共用每分钟1200限制**。
 - 每个请求将包含一个 `X-MBX-US-ED-WEIGHT-(intervalNum)(intervalLetter)` 的头，包含当前IP所有请求的已使用权重。
 - 每个成功的下单回报将包含一个 `X-MBX-ORDER-COUNT-(intervalNum)(intervalLetter)` 的头，其中包含当前账户已用的下单限制数量。
- `/sapi/*` 的接口相关：
 - 按IP和按UID(account)两种模式分别统计, 两者互相独立。
 - 以 `/sapi/*` 开头的接口采用**单接口限频模式**。按IP统计的权重单接口权重总额为每分钟12000；按照UID统计的单接口权重总额是每分钟180000。
 - 每个接口会标明是按照IP或者按照UID统计, 以及相应请求一次的权重值。
 - 按照IP统计的接口, 请求返回头里面会包含 `X-SAPI-US-ED-IP-WEIGHT-1M=<value>`, 包含当前IP所有请求已使用权重。
 - 按照UID统计的接口, 请求返回头里面会包含 `X-SAPI-US-ED-UID-WEIGHT-1M=<value>`, 包含当前账户所有已用的UID权重。

数据来源

- 因为API系统是异步的，所以返回的数据有延时很正常，也在预期之中。
- 在每个接口中，列出了其数据的来源，可以用于理解数据的时效性。

系统一共有3个数据来源，按照更新速度的先后排序。排在前面的数据最新，在后面就有可能存在延迟。

- **撮合引擎** - 表示数据来源于撮合引擎
- **缓存** - 表示数据来源于内部或者外部的缓存
- **数据库** - 表示数据直接来源于数据库

- ①** 有些接口有不止一个数据源，比如 `缓存 => 数据库`，这表示接口会先从第一个数据源检查，如果没有数据，则检查下一个数据源。

接口鉴权类型

- 每个接口都有自己的鉴权类型，鉴权类型决定了访问时应当进行何种鉴权。
- 鉴权类型会在本文档中各个接口名称旁声明，如果没有特殊声明即默认为 `NONE`。
- 如果需要 API-keys，应当在HTTP头中以 `X-MBX-APIKEY` 字段传递。
- API-keys 与 secret-keys 是**大小写敏感的**。
- API-keys 可以被配置为只拥有访问一些接口的权限。例如，一个 API-key 仅可用于发送交易指令，而另一个 API-key 则可访问除交易指令外的所有路径。
- 默认 API-keys 可访问所有鉴权路径。

鉴权类型	描述
<code>NONE</code>	不需要鉴权的接口
<code>TRADE</code>	需要有效的 API-Key 和签名
<code>MARGIN</code>	需要有效的 API-Key 和签名
<code>USER_DATA</code>	需要有效的 API-Key 和签名
<code>USER_STREAM</code>	需要有效的 API-Key
<code>MARKET_DATA</code>	需要有效的 API-Key

- `TRADE`, `MARGIN` 和 `USER_DATA` 接口是 签名(SIGNED)接口。

SIGNED (TRADE、USER_DATA AND MARGIN) Endpoint security

- 调用 `SIGNED` 接口时，除了接口本身所需的参数外，还需要在 `query string` 或 `request body` 中传递 `signature`，即签名参数。
- 签名使用 `HMAC SHA256` 算法。API-KEY所对应的API-Secret作为 `HMAC SHA256` 的密钥，其他所有参数作为 `HMAC SHA256` 的操作对象，得到的输出即为签名。
- 签名**大小写不敏感**。
- "totalParams" 定义为与 "request body" 串联的 "query string"。

时间同步安全

- 签名接口均需要传递 `timestamp` 参数，其值应当是请求发送时刻的 unix 时间戳(毫秒)。
- 服务器收到请求时会判断请求中的时间戳，如果是 5000 毫秒之前发出的，则请求会被认为无效。这个时间空窗值可以通过发送可选参数 `recvWindow` 来定义。

逻辑伪代码如下：

```

if (timestamp < (serverTime + 1000) && (serverTime - timestamp) <= recvWindow)
{
    // process request
}
else
{
    // reject request
}

```

关于交易时效性 互联网状况并不完全稳定可靠，因此你的程序本地到币安服务器的时延会有抖动。这是我们设置 `recvWindow` 的目的所在，如果你从事高频交易，对交易时效性有较高的要求，可以灵活设置 `recvWindow` 以达到你的要求。

- ① 推荐使用 5 秒以下的 `recvWindow`！最多不能超过 60 秒！

POST /API/V3/ORDER 的示例

以下是在 linux bash 环境下使用 echo openssl 和 curl 工具实现的一个调用接口下单的示例 apikey、secret 仅供示范

Key	Value
apiKey	vmPUZE6mv9SD5VNHk4HIWFsOr6aKE2zvsw0MuIgwCIPy6utIco14y7Ju91duEh8A
secretKey	NhqPtmdSjYdKjVHjA7PZj4Mge3R5YNiP1e3UZjInCIVN65XAbvqqM6A7H5fATj0j

参数	取值
symbol	LTCBTC
side	BUY
type	LIMIT
timeInForce	GTC
quantity	1
price	0.1
recvWindow	5000
timestamp	1499827319559

示例 1：所有参数通过 REQUEST BODY 发送

Example 1

HMAC SHA256 signature:

```
$ echo -n "symbol=LTCBTC&side=BUY&type=LIMIT&timeInForce=GTC&quantity=1&price=0.1&recvWindow=5000&timestamp=1499827319559" | openssl dgst -sha256 -hmac $secretKey
```

curl command:

```
(HMAC SHA256)
$ curl -H "X-MBX-APIKEY: vmPUZE6mv9SD5VNHk4HlWFsOr6aKE2zvsw0MuIgwCIPy6utIco14y7Ju91duEh8A" -X POST 'http://api.binance.com/api/v3/order'
```

- **requestBody:**

symbol=LTCBTC
 &side=BUY
 &type=LIMIT
 &timeInForce=GTC
 &quantity=1
 &price=0.1
 &recvWindow=5000
 ×tam

示例 2: 所有参数通过 QUERY STRING 发送**Example 2****HMAC SHA256 signature:**

```
$ echo -n "symbol=LTCBTC&side=BUY&type=LIMIT&timeInForce=GTC&quantity=1&price=0.1&recvWindow=5000&timestam"
(stdin)= c8db56825ae71d6d79447849e617115f4a920fa2acdab2b053c4b2838bd6b71
```

curl command:

```
(HMAC SHA256)
$ curl -H "X-MBX-APIKEY: vmPUZE6mv9SD5VNHk4HlWFsOr6aKE2zvsw0MuIgwCIPy6utIco14y7Ju91duEh8A" -X POST 'http://api.binance.com/api/v3/order'
```

- **queryString:**

symbol=LTCBTC
 &side=BUY
 &type=LIMIT
 &timeInForce=GTC
 &quantity=1
 &price=0.1
 &recvWindow=5000
 ×tam

示例 3: 混合使用 QUERY STRING 和 REQUEST BODY**Example 3****HMAC SHA256 signature:**

```
$ echo -n "symbol=LTCBTC&side=BUY&type=LIMIT&timeInForce=GTC&quantity=1&price=0.1&recvWindow=5000&timestam"
(stdin)= 0fd168b8ddb4876a0358a8d14d0c9f3da0e9b20c5d52b2a00fcf7d1c602f9a77
```

curl command:

```
(HMAC SHA256)
$ curl -H "X-MBX-APIKEY: vmPUZE6mv9SD5VNHk4H1WFsOr6aKE2zvsw0MuIgwCIPy6utIco14y7Ju91duEh8A" -X POST 'ht
```

- **queryString:**

symbol=LTCBTC&side=BUY&type=LIMIT&timeInForce=GTC

- **requestBody:**

quantity=1&price=0.1&recvWindow=5000×tamp=1499827319559

请注意，签名与示例3不同。 "GTC"和"quantity = 1"之间没有&。

公开 API 参数

术语

这里的术语适用于全部文档，建议特别是新手熟读，也便于理解。

- **base asset** 指一个交易对的交易对象，即写在靠前部分的资产名，比如BTCUSDT，BTC是**base asset**。
- **quote asset** 指一个交易对的定价资产，即写在靠后部分的资产名，比如BTCUSDT，USDT是**quote asset**。

枚举定义

交易对状态 (状态 status):

- PRE_TRADING 交易前
- TRADING 交易中
- POST_TRADING 交易后
- END_OF_DAY
- HALT
- AUCTION_MATCH
- BREAK

交易对类型:

- SPOT 现货
- MARGIN 杠杆
- LEVERAGED 杠杆代币
- TRD_GRP_002 交易组 002
- TRD_GRP_003 交易组 003
- TRD_GRP_004 交易组 004
- TRD_GRP_005 交易组 005

订单状态 (状态 status):

状态	描述
NEW	订单被交易引擎接受
PARTIALLY_FILLED	部分订单被成交
FILLED	订单完全成交
CANCELED	用户撤销了订单

状态	描述
PENDING_CANCEL	撤销中(目前并未使用)
REJECTED	订单没有被交易引擎接受，也没被处理
EXPIRED	订单被交易引擎取消, 比如 LIMIT FOK 订单没有成交 市价单没有完全成交 强平期间被取消的订单 交易所维护期间被取消的订单

OCO 状态 (状态类型集 listStatusType):

状态	描述
RESPONSE	当ListStatus响应失败的操作时使用。(订单完成或取消订单)
EXEC_STARTED	当已经下单或者订单有更新时
ALL_DONE	当订单执行结束或者不在激活状态

OCO 订单状态 (订单状态集 listOrderStatus):

状态	描述
EXECUTING	当已经下单或者订单有更新时
ALL_DONE	当订单执行结束或者不在激活状态
REJECT	当订单状态响应失败(订单完成或取消订单)

指定订单的类型

- OCO 选择性委托订单

订单类型 (orderTypes, type):

- LIMIT 限价单
- MARKET 市价单
- STOP_LOSS 止损单
- STOP_LOSS_LIMIT 限价止损单
- TAKE_PROFIT 止盈单
- TAKE_PROFIT_LIMIT 限价止盈单
- LIMIT_MAKER 限价只挂单

订单返回类型 (newOrderRespType):

- ACK
- RESULT
- FULL

订单方向 (方向 side):

- BUY 买入
- SELL 卖出

有效方式 (timeInForce):

这里定义了订单多久能够失效

Status	Description
--------	-------------

Status	Description
GTC	成交为止 订单会一直有效，直到被成交或者取消。
IOC	无法立即成交的部分就撤销 订单在失效前会尽量多的成交。
FOK	无法全部立即成交就撤销 如果无法全部成交，订单会失效。

K线间隔:

s -> 秒; m -> 分钟; h -> 小时; d -> 天; w -> 周; M -> 月

- 1s
- 1m
- 3m
- 5m
- 15m
- 30m
- 1h
- 2h
- 4h
- 6h
- 8h
- 12h
- 1d
- 3d
- 1w
- 1M

限制种类 (rateLimitType)

REQUEST_WEIGHT

```
{
  "rateLimitType": "REQUEST_WEIGHT",
  "interval": "MINUTE",
  "intervalNum": 1,
  "limit": 1200
}
```

ORDERS

```
{
  "rateLimitType": "ORDERS",
  "interval": "SECOND",
  "intervalNum": 10,
  "limit": 100
},
{
  "rateLimitType": "ORDERS",
  "interval": "DAY",
  "intervalNum": 1,
  "limit": 200000
}
```

RAW_REQUESTS

```
{
  "rateLimitType": "RAW_REQUESTS",
  "interval": "MINUTE",
  "intervalNum": 5,
  "limit": 5000
}
```

- REQUEST_WEIGHT 单位时间请求权重之和上限
- ORDERS 单位时间下单次数限制
- RAW_REQUESTS 单位时间请求次数上限

限制间隔 (interval)

- SECOND 秒
- MINUTE 分
- DAY 天

过滤器

过滤器，即Filter，定义了一系列交易规则。共有两类，分别是针对交易对的过滤器symbol filters，和针对整个交易所的过滤器exchange filters

交易对过滤器

PRICE_FILTER 价格过滤器

/exchangeInfo 响应中的格式:

```
{
  "filterType": "PRICE_FILTER",
  "minPrice": "0.00000100",
  "maxPrice": "100000.00000000",
  "tickSize": "0.00000100"
}
```

价格过滤器 用于检测订单中 price 参数的合法性。包含以下三个部分:

- minPrice 定义了 price/stopPrice 允许的最小值。
- maxPrice 定义了 price/stopPrice 允许的最大值。
- tickSize 定义了 price/stopPrice 的步进间隔，即price必须等于minPrice+(tickSize的整数倍)

以上每一项均可为0，为0时代表这一项不再做限制。

逻辑伪代码如下:

- price >= minPrice
- price <= maxPrice
- price % tickSize == 0

PERCENT_PRICE 价格振幅过滤器

/exchangeInfo 响应中的格式:

```
{
  "filterType": "PERCENT_PRICE",
  "multiplierUp": "5",
  "multiplierDown": "0.2",
  "avgPriceMins": 5
}
```

`PERCENT_PRICE` 过滤器基于先前交易的平均值来定义价格的有效范围。

`avgPriceMins` 是计算平均价格的分钟数。 0 表示使用最后的价格。

为了通过“价格百分比”，“价格”必须符合以下条件：

- `price <= weightedAveragePrice * multiplierUp`
- `price >= weightedAveragePrice * multiplierDown`

PERCENT_PRICE_BY_SIDE 基于买卖方向的价格振幅过滤器

ExchangeInfo format:

```
{
  "filterType": "PERCENT_PRICE_BY_SIDE",
  "bidMultiplierUp": "1.2",
  "bidMultiplierDown": "0.2",
  "askMultiplierUp": "5",
  "askMultiplierDown": "0.8",
  "avgPriceMins": 1
}
```

`PERCENT_PRICE_BY_SIDE` 过滤器定义了基于交易对平均价格的合法价格范围。取决于 `BUY` 或者 `SELL`，价格范围可能有所不同。

`avgPriceMins` 是用来计算平均价格的分钟数。0 表示用最新价(last price)。

买向订单需要满足：

- `Order price <= weightedAveragePrice * bidMultiplierUp`
- `Order price >= weightedAveragePrice * bidMultiplierDown`

卖向订单需要满足：

- `Order Price <= weightedAveragePrice * askMultiplierUp`
- `Order Price >= weightedAveragePrice * askMultiplierDown`

LOT_SIZE 订单尺寸

/exchangeInfo 响应中的格式：

```
{
  "filterType": "LOT_SIZE",
  "minQty": "0.00100000",
  "maxQty": "100000.00000000",
  "stepSize": "0.00100000"
}
```

Lots 是拍卖术语，`LOT_SIZE` 过滤器对订单中的 `quantity` 也就是数量参数进行合法性检查。包含三个部分：

- `minQty` 表示 `quantity / icebergQty` 允许的最小值。
- `maxQty` 表示 `quantity / icebergQty` 允许的最大值。
- `stepSize` 表示 `quantity / icebergQty` 允许的步进值。

逻辑伪代码如下:

- `quantity` \geq `minQty`
- `quantity` \leq `maxQty`
- $(\text{quantity} - \text{minQty}) \% \text{stepSize} = 0$

MIN_NOTIONAL 最小名义价值(成交额)

/exchangeInfo 响应中的格式:

```
{
  "filterType": "MIN_NOTIONAL",
  "minNotional": "0.00100000",
  "applyToMarket": true,
  "avgPriceMins": 5
}
```

MIN_NOTIONAL过滤器定义了交易对订单所允许的最小名义价值(成交额)。订单的名义价值是`价格 * 数量`。如果是高级订单(比如止损止损订单`STOP_LOSS_LIMIT`)，名义价值会按照`stopPrice * quantity`来计算。如果是冰山订单，名义价值会按照`price * icebergQty`来计算。`applyToMarket`确定`MIN_NOTIONAL`过滤器是否也将应用于`MARKET`订单。由于`MARKET`订单没有价格，因此会在最后`avgPriceMins`分钟内使用平均价格。`avgPriceMins`是计算平均价格的分钟数。0表示使用最后的价格。

NOTIONAL 名义价值

/exchangeInfo 响应中的格式:

```
{
  "filterType": "NOTIONAL",
  "minNotional": "10.00000000",
  "applyMinToMarket": false,
  "maxNotional": "10000.00000000",
  "applyMaxToMarket": false,
  "avgPriceMins": 5
}
```

名义价值过滤器(NOTIONAL)定义了订单在一个交易对上可以下单的名义价值区间。

`applyMinToMarket` 定义了 `minNotional` 是否适用于市价单(`MARKET`)
`applyMaxToMarket` 定义了 `maxNotional` 是否适用于市价单(`MARKET`)。

要通过此过滤器，订单的名义价值(单价 \times 数量, `price * quantity`)需要满足如下条件:

- `price * quantity` \leq `maxNotional`
- `price * quantity` \geq `minNotional`

对于市价单(`MARKET`)，用于计算的价格采用的是在`avgPriceMins` 定义的时间之内的平均价。
如果`avgPriceMins` 为 0，则采用最新的价格。

ICEBERG_PARTS 冰山订单拆分数

/exchangeInfo 响应中的格式:

```
{
  "filterType": "ICEBERG_PARTS",
  "limit": 10
}
```

`ICEBERG_PARTS` 代表冰山订单最多可以拆分成多少个小订单。
计算方法为 向上取整(`qty / icebergQty`)。

MARKET_LOT_SIZE 市价订单尺寸

/exchangeInfo 响应中的格式

```
{
  "filterType": "MARKET_LOT_SIZE",
  "minQty": "0.00100000",
  "maxQty": "100000.00000000",
  "stepSize": "0.00100000"
}
```

`MARKET_LOT_SIZE` 过滤器为交易对上的 `MARKET` 订单定义了 `数量`(即拍卖中的"手数")规则。 共有3部分：

- `minQty` 定义了允许的最小 `quantity`。
- `maxQty` 定义了允许的最大数量。
- `stepSize` 定义了可以增加/减少数量的间隔。

为了通过 `market lot size` , `quantity` 必须满足以下条件：

- `quantity >= minQty`
- `quantity <= maxQty`
- `(quantity - minQty) % stepSize == 0`

MAX_NUM_ORDERS 最多订单数

/exchangeInfo 响应中的格式:

```
{
  "filterType": "MAX_NUM_ORDERS",
  "maxNumOrders": 25
}
```

定义了某个交易对最多允许的挂单数量(不包括已关闭的订单)

普通订单与条件订单均计算在内

MAX_NUM_ALGO_ORDERS 最多条件单数

/exchangeInfo 响应中的格式:

```
{
  "filterType": "MAX_NUM_ALGO_ORDERS",
  "maxNumAlgoOrders": 5
}
```

`MAX_NUM_ALGO_ORDERS` 过滤器定义允许账户在交易对上开设的"algo"订单的最大数量。

"Algo"订单是 `STOP_LOSS` , `STOP_LOSS_LIMIT` , `TAKE_PROFIT` 和 `TAKE_PROFIT_LIMIT` 止盈止损单。

MAX_NUM_ICEBERG_ORDERS 最多冰山单数

`MAX_NUM_ICEBERG_ORDERS` 过滤器定义了允许在交易对上开设账户的 `ICEBERG` 订单的最大数量。

`ICEBERG` 订单是 `icebergQty` 大于0的任何订单。 .

/exchangeInfo 响应中的格式:

```
{
  "filterType": "MAX_NUM_ICEBERG_ORDERS",
  "maxNumIcebergOrders": 5
}
```

MAX_POSITION 过滤器

这个过滤器定义账户允许的基于 `base asset` 的最大仓位。一个用户的仓位可以定义为如下资产的总和: 1. `base asset` 的可用余额 1. `base asset` 的锁定余额 1. 所有处于 `open` 的买单的数量总和

如果用户的仓位大于最大的允许仓位，买单会被拒绝。

如果一个订单的数量(`quantity`) 可能导致持有仓位溢出，会触发过滤器 `MAX_POSITION`。

`/exchangeInfo` 响应中的格式:

```
{
  "filterType": "MAX_POSITION",
  "maxPosition": "10.00000000"
}
```

TRAILING_DELTA

`ExchangeInfo` 格式:

```
{
  "filterType": "TRAILING_DELTA",
  "minTrailingAboveDelta": 10,
  "maxTrailingAboveDelta": 2000,
  "minTrailingBelowDelta": 10,
  "maxTrailingBelowDelta": 2000
}
```

此过滤器定义了参数 `trailingDelta` 的最大和最小值。

下追踪止损订单，需要满足条件:

对于 `STOP_LOSS BUY`, `STOP_LOSS_LIMIT_BUY`, `TAKE_PROFIT_SELL` 和 `TAKE_PROFIT_LIMIT_SELL` 订单:

- `trailingDelta` \geq `minTrailingAboveDelta`
- `trailingDelta` \leq `maxTrailingAboveDelta`

对于 `STOP_LOSS_SELL`, `STOP_LOSS_LIMIT_SELL`, `TAKE_PROFIT_BUY` 和 `TAKE_PROFIT_LIMIT_BUY` 订单:

- `trailingDelta` \geq `minTrailingBelowDelta`
- `trailingDelta` \leq `maxTrailingBelowDelta`

交易所级别过滤器

EXCHANGE_MAX_NUM_ORDERS 最多订单数

`/exchangeInfo` 响应中的格式:

```
{
  "filterType": "EXCHANGE_MAX_NUM_ORDERS",
  "maxNumOrders": 1000
}
```

EXCHANGE_MAX_NUM_ORDERS 过滤器定义了允许在交易对上开设账户的最大订单数。

请注意，此过滤器同时计算"algo"订单和常规订单。

EXCHANGE_MAX_ALGO_ORDERS 交易最大ALGO订单数

/exchangeInfo 响应中的格式:

```
{
  "filterType": "EXCHANGE_MAX_ALGO_ORDERS",
  "maxNumAlgoOrders": 200
}
```

EXCHANGE_MAX_ALGO_ORDERS 过滤器定义了允许在交易上开设账户的"algo"订单的最大数量。

"Algo"订单是 **STOP_LOSS** , **STOP_LOSS_LIMIT** , **TAKE_PROFIT** 和 **TAKE_PROFIT_LIMIT** 订单。

EXCHANGE_MAX_NUM_ICEBERG_ORDERS 冰山订单的最大订单数

此过滤器定义了允许账号持有的最大冰山订单数量。

/exchangeInfo 响应中的格式:

```
{
  "filterType": "EXCHANGE_MAX_NUM_ICEBERG_ORDERS",
  "maxNumIcebergOrders": 10000
}
```

钱包接口

系统状态(System)

响应

```
{
  "status": 0,          // 0: 正常, 1: 系统维护
  "msg": "normal"      // "normal", "system_maintenance"
}
```

GET /sapi/v1/system/status

获取系统状态。

权重(IP): 1

获取所有币信息 (USER_DATA)

获取针对用户的所有(Binance支持充提操作)的币种信息。

响应

```
[
  {
    "coin": "BTC",
    "depositAllEnable": true,
    "free": "0.08074558",
    "freeze": "0.00000000",
    "ipoable": "0.00000000",
    "ipoing": "0.00000000",
    "isLegalMoney": false,
    "locked": "0.00000000",
    "name": "Bitcoin",
    "networkList": [
      {
        "addressRegex": "^(bnb1)[0-9a-z]{38}$",
        "coin": "BTC",
        "depositDesc": "Wallet Maintenance, Deposit Suspended", // 仅在充值关闭时返回
        "depositEnable": false,
        "isDefault": false,
        "memoRegex": "^[0-9A-Za-z\\-_]{1,120}$",
        "minConfirm": 1, // 上账所需的最小确认数
        "name": "BEP2",
        "network": "BNB",
        "resetAddressStatus": false,
        "specialTips": "Both a MEMO and an Address are required to successfully deposit your BEP2-BTC",
        "unLockConfirm": 0, // 解锁需要的确认数
        "withdrawDesc": "Wallet Maintenance, Withdrawal Suspended", // 仅在提现关闭时返回
        "withdrawEnable": false,
        "withdrawFee": "0.00000220",
        "withdrawIntegerMultiple": "0.00000001",
        "withdrawMax": "9999999999.99999999",
        "withdrawMin": "0.00000440",
        "sameAddress": true, // 是否需要memo
        "estimatedArrivalTime": 25,
        "busy": false
      },
      {
        "addressRegex": "^[13][a-km-zA-HJ-NP-Z1-9]{25,34}$|^bc1[0-9A-Za-z]{39,59}$",
        "coin": "BTC",
        "depositEnable": true,
        "isDefault": true,
        "memoRegex": "",
        "minConfirm": 1, // 上账所需的最小确认数
        "name": "BTC",
        "network": "BTC",
        "resetAddressStatus": false,
        "specialTips": "",
        "unLockConfirm": 0, // 解锁需要的确认数
        "withdrawEnable": true,
        "withdrawFee": "0.00050000",
        "withdrawIntegerMultiple": "0.00000001",
        "withdrawMax": "750",
        "withdrawMin": "0.00100000",
        "sameAddress": false,
        "estimatedArrivalTime": 25,
        "busy": false
      }
    ],
  ]
]
```

```

    "storage": "0.00000000",
    "trading": true,
    "withdrawAllEnable": true,
    "withdrawning": "0.00000000"
}
]

```

`GET /sapi/v1/capital/config/getall (HMAC SHA256)`

权重(IP): 10

参数:

名称	类型	是否必需	描述
recvWindow	LONG	NO	
timestamp	LONG	YES	

查询每日资产快照 (USER_DATA)

响应

```
{
  "code":200, // 200表示返回正确, 否则即为错误码
  "msg":"",
  "snapshotVos":[
    {
      "data":{
        "balances":[
          {
            "asset":"BTC",
            "free":"0.09905021",
            "locked":"0.00000000"
          },
          {
            "asset":"USDT",
            "free":"1.89109409",
            "locked":"0.00000000"
          }
        ],
        "totalAssetOfBtc":"0.09942700"
      },
      "type":"spot",
      "updateTime":1576281599000
    }
  ]
}
```

或

```
{
  "code":200, // 200表示返回正确，否则即为错误码
  "msg":"",
  "snapshotVos":[
    {
      "data":{
        "marginLevel":"2748.02909813",
        "totalAssetOfBtc":"0.00274803",
        "totalLiabilityOfBtc":"0.00000100",
        "totalNetAssetOfBtc":"0.00274750",
        "userAssets":[
          {
            "asset":"XRP",
            "borrowed":"0.00000000",
            "free":"1.00000000",
            "interest":"0.00000000",
            "locked":"0.00000000",
            "netAsset":"1.00000000"
          }
        ]
      },
      "type":"margin",
      "updateTime":1576281599000
    }
  ]
}
```

或

```
{
  "code":200, // 200表示返回正确，否则即为错误码
  "msg":"",
  "snapshotVos":[
    {
      "data":{
        "assets":[
          {
            "asset":"USDT",
            "marginBalance":"118.99782335",
            "walletBalance":"120.23811389"
          }
        ],
        "position":[
          {
            "entryPrice":"7130.41000000",
            "markPrice":"7257.66239673",
            "positionAmt":"0.01000000",
            "symbol":"BTCUSDT",
            "unRealizedProfit":"1.24029054" //只显示开仓当时的未实现盈亏，不会实时更新，可以忽略
          }
        ]
      },
      "type":"futures",
      "updateTime":1576281599000
    }
}
```

]
}

GET /sapi/v1/accountSnapshot (HMAC SHA256)

权重(IP): 2400**参数:**

名称	类型	是否必需	描述
type	STRING	YES	"SPOT", "MARGIN", "FUTURES"
startTime	LONG	NO	
endTime	LONG	NO	
limit	INT	NO	min 7, max 30, default 7
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 查询时间范围最大不得超过30天
- 仅支持查询最近 1 个月数据
- 若 startTime 和 endTime 没传，则默认返回最近7天数据

关闭站内划转 (USER_DATA)

响应

{}

POST /sapi/v1/account/disableFastWithdrawSwitch (HMAC SHA256)

权重(IP): 1**参数:**

名称	类型	是否必需	描述
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 注意:

此请求会关闭您账户的站内快速划转。您需要为 api-key 开通 "trade" 权限才能发送此请求。

开启站内划转 (USER_DATA)

响应

{}

POST /sapi/v1/account/enableFastWithdrawSwitch (HMAC SHA256)

权重(IP): 1**参数:**

名称	类型	是否必需	描述
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 此请求会开启您账户的站内快速划转。您需要为api-key开通"trade"权限才能发送此请求。
- 开启以后, 如果收款方为币安账户地址, 转账费用为0, 速度快, 不需要提交上链请求。

提币 (USER_DATA)

响应

```
{
  "id": "7213fea8e94b4a5593d507237e5a555b"
}
```

POST /sapi/v1/capital/withdraw/apply (HMAC SHA256)

Submit a withdraw request.

权重(UID): 600**参数:**

名称	类型	是否必需	描述
coin	STRING	YES	
withdrawOrderId	STRING	NO	自定义提币ID
network	STRING	NO	提币网络
address	STRING	YES	提币地址
addressTag	STRING	NO	某些币种例如 XRP,XMR 允许填写次级地址标签
amount	DECIMAL	YES	数量
transactionFeeFlag	BOOLEAN	NO	当站内转账时免手续费, <code>true</code> : 手续费归资金转入方; <code>false</code> : 手续费归资金转出方; . 默认 <code>false</code> .
name	STRING	NO	地址的备注, 填写该参数后会加入该币种的提现地址簿。地址簿上限为20, 超出后会造成提现失败。地址中的空格需要encode成%20
walletType	INTEGER	NO	表示出金使用的钱包, 0为现货钱包, 1为资金钱包, 默认为现货钱包

名称	类型	是否必需	描述
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 如果不发送 `network`, 将按该币种默认网络返回结果 ;
- 可以在接口 `Get /sapi/v1/capital/config/getall (HMAC SHA256)` 的返回值中某币种的 `networkList` 获取 `network` 网络字段和 `isDefault` 是否为默认网络。

获取充值历史(支持多网络) (USER_DATA)

响应

```
[
  {
    "id": "769800519366885376",
    "amount": "0.001",
    "coin": "BNB",
    "network": "BNB",
    "status": 0,
    "address": "bnb136ns6lfw4zs5hg4n85vdthaad7hq5m4gtkgf23",
    "addressTag": "101764890",
    "txId": "98A3EA560C6B3336D348B6C83F0F95ECE4F1F5919E94BD006E5BF3BF264FACFC",
    "insertTime": 1661493146000,
    "transferType": 0,
    "confirmTimes": "1/1",
    "unlockConfirm": 0,
    "walletType": 0
  },
  {
    "id": "769754833590042625",
    "amount": "0.50000000",
    "coin": "IOTA",
    "network": "IOTA",
    "status": 1,
    "address": "SIZ9VLMHWATXKV99LH99CIGFJFUMLEHGWVZVNNZXRRJJWBPHWPPBOSDORZ9EQSHCZAMPVAPGFYQAUUV9DROOXJL",
    "addressTag": "",
    "txId": "ESBFVQUTPIWNJSNXFHNYHSQNTGKRVKPRABQWTAXCDWOAKDKYWPTVG9BGXNVNKTLEJGESAVXIKIZ9999",
    "insertTime": 1599620082000,
    "transferType": 0,
    "confirmTimes": "1/1",
    "unlockConfirm": 0,
    "walletType": 0
  }
]
```

`GET /sapi/v1/capital/deposit/hisrec (HMAC SHA256)`

权重(IP): 1

参数:

名称	类型	是否必需	描述
coin	STRING	NO	
status	INT	NO	0(0:pending,6: credited but cannot withdraw, 1:success)
startTime	LONG	NO	默认当前时间90天前的时间戳
endTime	LONG	NO	默认当前时间戳
offset	INT	NO	默认:0
limit	INT	NO	默认：1000，最大1000
recvWindow	LONG	NO	
timestamp	LONG	YES	
txId	STRING	NO	

- 请注意 `startTime` 与 `endTime` 的默认时间戳，保证请求时间间隔不超过90天.
- 同时提交 `startTime` 与 `endTime` 间隔不得超过90天.

获取提币历史 (支持多网络) (USER_DATA)

响应

```
[
  {
    "address": "0x94df8b352de7f46f64b01d3666bf6e936e44ce60",
    "amount": "8.91000000", // 提现转出金额
    "applyTime": "2019-10-12 11:12:02", // UTC 时间
    "coin": "USDT",
    "id": "b6ae22b3aa844210a7041aee7589627c", // 该笔提现现在币安的id
    "withdrawOrderId": "WITHDRAWtest123", // 自定义ID，如果没有则不返回该字段
    "network": "ETH",
    "transferType": 0 // 1: 站内转账，0: 站外转账
    "status": 6,
    "transactionFee": "0.004", // 手续费
    "confirmNo": 3, // 提现确认数
    "info": "The address is not valid. Please confirm with the recipient", // 提币失败原因
    "txId": "0xb5ef8c13b968a406cc62a93a8bd80f9e9a906ef1b3fcf20a2e48573c17659268" // 提现交易id
  },
  {
    "address": "1FZdVHtiBqMrWdjPyRPULCUceZPJ2WLCsB",
    "amount": "0.00150000",
    "applyTime": "2019-09-24 12:43:45",
    "coin": "BTC",
    "id": "156ec387f49b41df8724fa744fa82719",
    "network": "BTC",
    "transferType": 0, // 1: 站内转账，0: 站外转账
    "status": 6,
    "transactionFee": "0.004",
    "confirmNo": 2,
    "info": ""
  }
]
```

```

        "txId": "60fd9007ebfddc753455f95faf808c4302c836e4d1eebc5a132c36c1d8ac354"
    }
]

```

`GET /sapi/v1/capital/withdraw/history (HMAC SHA256)`

权重(IP): 1

参数:

名称	类型	是否必需	描述
coin	STRING	NO	
withdrawOrderId	STRING	NO	
status	INT	NO	0(0:已发送确认Email,1:已被用户取消 2:等待确认 3:被拒绝 4:处理中 5:提现交易失败 6 提现完成)
offset	INT	NO	
limit	INT	NO	默认 : 1000 , 最大 : 1000
startTime	LONG	NO	默认当前时间90天前的时间戳
endTime	LONG	NO	默认当前时间戳
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 支持多网络提币前的历史记录可能不会返回 `network` 字段.
- 请注意 `startTime` 与 `endTime` 的默认时间戳，保证请求时间间隔不得超过90天.
- 同时提交 `startTime` 与 `endTime` 间隔不得超过90天.
- 若传了 `withdrawOrderId`，则请求的 `startTime` 与 `endTime` 的时间间隔不得超过7天.
- 若传了 `withdrawOrderId`，没传 `startTime` 与 `endTime`，则默认返回最近7天数据.

获取充值地址 (支持多网络) (USER_DATA)

响应

```
{
    "address": "1HPn8Rx2y6nNSfagQbKy27GB99Vbzg89wv",
    "coin": "BTC",
    "tag": "",
    "url": "https://btc.com/1HPn8Rx2y6nNSfagQbKy27GB99Vbzg89wv"
}
```

`GET /sapi/v1/capital/deposit/address (HMAC SHA256)`

权重(IP): 10

参数:

名称	类型	是否必需	描述
----	----	------	----

名称	类型	是否必需	描述
coin	STRING	YES	
network	STRING	NO	
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 如果不发送 `network`, 将按该币种默认网络返回结果;
- 可以在接口 `Get /sapi/v1/capital/config/getall (HMAC SHA256)` 的返回值中某币种的 `networkList` 获取 `network` 网络字段和 `isDefault` 是否为默认网络。

账户状态 (USER_DATA)

响应

```
{
  "data": "Normal"
}
```

`GET /sapi/v1/account/status`

获取账户状态详情。

权重(IP): 1

参数:

名称	类型	是否必需	描述
recvWindow	LONG	NO	
timestamp	LONG	YES	

账户API交易状态(USER_DATA)

响应

```
{
  "data": {
    // 账户API交易状态详情
    "isLocked": false, // API交易功能是否被锁
    "plannedRecoverTime": 0, // API交易功能被锁情况下的预计恢复时间
    "triggerCondition": {
      "GCR": 150, // Number of GTC orders
      "IFER": 150, // Number of FOK/IOC orders
      "UFR": 300 // Number of orders
    },
    "updateTime": 1547630471725
  }
}
```

}

GET /sapi/v1/account/apiTradingStatus (HMAC SHA256)

获取 api 账户交易状态详情。

权重(IP): 1

参数:

名称	类型	是否必需	描述
recvWindow	LONG	NO	
timestamp	LONG	YES	

小额资产转换BNB历史 (USER_DATA)

响应

```
{
    "total": 8,      //共计发生过的转换笔数
    "userAssetDribbles": [
        {
            "operateTime": 1615985535000,
            "totalTransferredAmount": "0.00132256",    //本次转换所得BNB
            "totalServiceChargeAmount": "0.00002699",    //本次转换手续费(BNB)
            "transId": 45178372831,
            "userAssetDribbleDetails": [                //本次转换的细节
                {
                    "transId": 4359321,
                    "serviceChargeAmount": "0.000009",
                    "amount": "0.0009",
                    "operateTime": 1615985535000,
                    "transferredAmount": "0.000441",
                    "fromAsset": "USDT"
                },
                {
                    "transId": 4359321,
                    "serviceChargeAmount": "0.00001799",
                    "amount": "0.0009",
                    "operateTime": 1615985535000,
                    "transferredAmount": "0.00088156",
                    "fromAsset": "ETH"
                }
            ]
        },
        {
            "operateTime": 1616203180000,
            "totalTransferredAmount": "0.00058795",
            "totalServiceChargeAmount": "0.000012",
            "transId": 4357015,
            "userAssetDribbleDetails": [

```

```
{
    "transId": 4357015,
    "serviceChargeAmount": "0.00001"
    "amount": "0.001",
    "operateTime": 1616203180000,
    "transferredAmount": "0.00049",
    "fromAsset": "USDT"
},
{
    "transId": 4357015,
    "serviceChargeAmount": "0.00002"
    "amount": "0.001",
    "operateTime": 1616203180000,
    "transferredAmount": "0.00009795",
    "fromAsset": "ETH"
}
]
}
]
}
}
```

`GET /sapi/v1/asset/dribblet (HMAC SHA256)`

权重(IP): 1

参数:

名称	类型	是否必需	描述
startTime	LONG	NO	
endTime	LONG	NO	
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 只返回最近100条记录
- 只返回 2020/12/01 之后记录

获取可以转换成BNB的小额资产 (USER_DATA)

响应

```
{
    "details": [
        {
            "asset": "ADA", //资产名
            "assetFullName": "ADA", //资产全称
            "amountFree": "6.21", //可转换数量
            "toBTC": "0.00016848", //等值BTC
            "toBNB": "0.01777302", //可转换BNB (未扣除手续费)
            "toBNBOffExchange": "0.01741756", //可转换BNB (已扣除手续费)
            "exchange": "0.00035546" //手续费
        }
    ]
}
```

```

    },
    [
      "totalTransferBtc": "0.00016848", //全部资产等值BTC
      "totalTransferBNB": "0.01777302", //总共可以转换的BNB数量
      "dribbletPercentage": "0.02"      //转换手续费
    }
  ]
}

```

POST /sapi/v1/asset/dust-btc (HMAC SHA256)

获取可以转换成 BNB 的小额资产列表.

权重(IP): 1

参数:

名称	类型	是否必需	描述
recvWindow	LONG	NO	
timestamp	LONG	YES	

小额资产转换 (USER_DATA)

响应

```
{
  "totalServiceCharge": "0.02102542",
  "totalTransferred": "1.05127099",
  "transferResult": [
    {
      "amount": "0.03000000",
      "fromAsset": "ETH",
      "operateTime": 1563368549307,
      "serviceChargeAmount": "0.00500000",
      "tranId": 2970932918,
      "transferredAmount": "0.25000000"
    },
    {
      "amount": "0.09000000",
      "fromAsset": "LTC",
      "operateTime": 1563368549404,
      "serviceChargeAmount": "0.01548000",
      "tranId": 2970932918,
      "transferredAmount": "0.77400000"
    },
    {
      "amount": "248.61878453",
      "fromAsset": "TRX",
      "operateTime": 1563368549489,
      "serviceChargeAmount": "0.00054542",
      "tranId": 2970932918,
      "transferredAmount": "0.02727099"
    }
  ]
}
```

```
[  
]
```

POST /sapi/v1/asset/dust (HMAC SHA256)

把小额资产转换成 BNB.

权重(UUID): 10

参数:

名称	类型	是否必需	描述
asset	ARRAY	YES	正在转换的资产。 例如 : asset = BTC & asset = USDT
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 您需要为API Key开通允许现货和杠杆交易权限才能发送此请求

资产利息记录 (USER_DATA)

响应

```
{
  "rows": [
    {
      "id": 1637366104,
      "amount": "10.0000000",
      "asset": "BHFT",
      "divTime": 1563189166000,
      "enInfo": "BHFT distribution",
      "tranId": 2968885920
    },
    {
      "id": 1631750237,
      "amount": "10.0000000",
      "asset": "BHFT",
      "divTime": 1563189165000,
      "enInfo": "BHFT distribution",
      "tranId": 2968885920
    }
  ],
  "total": 2
}
```

GET /sapi/v1/asset/assetDividend (HMAC SHA256)

获取资产利息记录。

权重(IP): 10

参数:

名称	类型	是否必需	描述
asset	STRING	NO	
startTime	LONG	NO	
endTime	LONG	NO	
limit	INT	NO	Default 20, max 500
recvWindow	LONG	NO	
timestamp	LONG	YES	

上架资产详情 (USER_DATA)

响应

```
{
  "CTR": {
    "minWithdrawAmount": "70.00000000",      //最小提现数量
    "depositStatus": false,      //是否可以充值(只有所有网络都关闭充值才为false)
    "withdrawFee": 35,      // 提现手续费
    "withdrawStatus": true,      //是否开放提现(只有所有网络都关闭提币才为false)
    "depositTip": "Delisted, Deposit Suspended"      //暂停充值的原因(如果暂停才有这一项)
  },
  "SKY": {
    "minWithdrawAmount": "0.02000000",
    "depositStatus": true,
    "withdrawFee": 0.01,
    "withdrawStatus": true
  }
}
```

GET /sapi/v1/asset/assetDetail (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
asset	STRING	NO	
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 充提币信息，建议查询 `GET /sapi/v1/capital/config/getall` 获取详情。

交易手续费率查询 (USER_DATA)

响应

```
[
  {
    "symbol": "ADABNB",
    "makerCommission": "0.001",
    "takerCommission": "0.001"
  },
  {
    "symbol": "BNBBTC",
    "makerCommission": "0.001",
    "takerCommission": "0.001"
  }
]
```

`GET /sapi/v1/asset/tradeFee (HMAC SHA256)`

权重(IP): 1

参数:

名称	类型	是否必需	描述
symbol	STRING	NO	
recvWindow	LONG	NO	
timestamp	LONG	YES	

用户万向划转 (USER_DATA)

响应:

```
{
  "tranId":13526853623
}
```

`POST /sapi/v1/asset/transfer (HMAC SHA256)`

您需要开通api key 允许万向划转权限来调用此接口。

权重(UUID): 900

参数:

名称	类型	是否必需	描述
type	ENUM	YES	
asset	STRING	YES	
amount	DECIMAL	YES	

名称	类型	是否必需	描述
fromSymbol	STRING	NO	
toSymbol	STRING	NO	
recvWindow	LONG	NO	
timestamp	LONG	YES	

- `fromSymbol` 必须要发送，当类型为 ISOLATEDMARGIN_MARGIN 和 ISOLATEDMARGIN_ISOLATEDMARGIN
- `toSymbol` 必须要发送，当类型为 MARGIN_ISOLATEDMARGIN 和 ISOLATEDMARGIN_ISOLATEDMARGIN
- 目前支持的type划转类型：
 - MAIN_UMFUTURE 现货钱包转向U本位合约钱包
 - MAIN_CMFUTURE 现货钱包转向币本位合约钱包
 - MAIN_MARGIN 现货钱包转向杠杆全仓钱包
 - UMFUTURE_MAIN U本位合约钱包转向现货钱包
 - UMFUTURE_MARGIN U本位合约钱包转向杠杆全仓钱包
 - CMFUTURE_MAIN 币本位合约钱包转向现货钱包
 - Margin_Main 杠杆全仓钱包转向现货钱包
 - Margin_UMFUTURE 杠杆全仓钱包转向U本位合约钱包
 - Margin_CMFUTURE 杠杆全仓钱包转向币本位合约钱包
 - CMFUTURE_MARGIN 币本位合约钱包转向杠杆全仓钱包
 - ISOLATEDMARGIN_MARGIN 杠杆逐仓钱包转向杠杆全仓钱包
 - Margin_ISOLATEDMARGIN 杠杆全仓钱包转向杠杆逐仓钱包
 - ISOLATEDMARGIN_ISOLATEDMARGIN 杠杆逐仓钱包转向杠杆逐仓钱包
 - MAIN_FUNDING 现货钱包转向资金钱包
 - FUNDING_MAIN 资金钱包转向现货钱包
 - FUNDING_UMFUTURE 资金钱包转向U本位合约钱包
 - UMFUTURE_FUNDING U本位合约钱包转向资金钱包
 - Margin_Funding 杠杆全仓钱包转向资金钱包
 - FUNDING_MARGIN 资金钱包转向杠杆全仓钱包
 - FUNDING_CMFUTURE 资金钱包转向币本位合约钱包
 - CMFUTURE_FUNDING 币本位合约钱包转向资金钱包

查询用户万向划转历史 (USER_DATA)

响应:

```
{
  "total":2,
  "rows":[
    {
      "asset":"USDT",
      "amount":"1",
      "type":"MAIN_UMFUTURE",
      "status": "CONFIRMED",
      "tranId": 11415955596,
      "timestamp":1544433328000
    },
    {
      "asset":"USDT",
      "amount":"2",
      "type":"MAIN_UMFUTURE",
      "status": "PENDING_CREATE"
    }
  ]
}
```

```

        "status": "CONFIRMED",
        "tranId": 11366865406,
        "timestamp":1544433328000
    }
]
}

```

`GET /sapi/v1/asset/transfer (HMAC SHA256)`

权重(IP): 1

参数:

名称	类型	是否必需	描述
type	ENUM	YES	
startTime	LONG	NO	
endTime	LONG	NO	
current	INT	NO	默认 1
size	INT	NO	默认 10, 最大 100
fromSymbol	STRING	NO	
toSymbol	STRING	NO	
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 若 `fromSymbol` 必须要发送，当类型为 ISOLATEDMARGIN_MARGIN 和 ISOLATEDMARGIN_ISOLATEDMARGIN
- 若 `toSymbol` 必须要发送，当类型为 MARGIN_ISOLATEDMARGIN 和 ISOLATEDMARGIN_ISOLATEDMARGIN
- 仅支持查询最近半年(6个月)数据
- 若 `startTime` 和 `endTime` 没传，则默认返回最近7天数据

资金账户 (USER_DATA)

响应

```

[
  {
    "asset": "USDT",
    "free": "1",      // 可用余额
    "locked": "0",   // 锁定资金
    "freeze": "0",   // 冻结资金
    "withdrawning": "0", // 提币
    "btcValuation": "0.00000091" // btc估值
  }
]

```

`POST /sapi/v1/asset/get-funding-asset (HMAC SHA256)`

权重(IP): 1

参数:

名称	类型	是否必需	描述
asset	STRING	NO	
needBtcValuation	STRING	NO	true or false
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 目前仅支持查询以下业务资产 : Binance Pay, Binance Card, Binance Gift Card, Stock Token

用户持仓 (USER_DATA)

响应

```
[
  {
    "asset": "AVAX",
    "free": "1",
    "locked": "0",
    "freeze": "0",
    "withdrawning": "0",
    "ipoable": "0",
    "btcValuation": "0"
  },
  {
    "asset": "BCH",
    "free": "0.9",
    "locked": "0",
    "freeze": "0",
    "withdrawning": "0",
    "ipoable": "0",
    "btcValuation": "0"
  },
  {
    "asset": "BNB",
    "free": "887.47061626",
    "locked": "0",
    "freeze": "10.52",
    "withdrawning": "0.1",
    "ipoable": "0",
    "btcValuation": "0"
  },
  {
    "asset": "BUSD",
    "free": "9999.7",
    "locked": "0",
    "freeze": "0",
    "withdrawning": "0",
    "ipoable": "0",
    "btcValuation": "0"
  }
]
```

```
{
  "asset": "SHIB",
  "free": "532.32",
  "locked": "0",
  "freeze": "0",
  "withdrawning": "0",
  "ipoable": "0",
  "btcValuation": "0"
},
{
  "asset": "USDT",
  "free": "50300000001.44911105",
  "locked": "0",
  "freeze": "0",
  "withdrawning": "0",
  "ipoable": "0",
  "btcValuation": "0"
},
{
  "asset": "WRZ",
  "free": "1",
  "locked": "0",
  "freeze": "0",
  "withdrawning": "0",
  "ipoable": "0",
  "btcValuation": "0"
}
]
```

POST /sapi/v3/asset/getUserAsset

获取用户持仓，仅返回>0的数据。

权重(IP): 5

参数:

名称	类型	是否必需	描述
asset	STRING	NO	如果资产为空，则查询用户所有的正资产。
needBtcValuation	BOOLEAN	NO	是否需要返回兑换成BTC的估值
recvWindow	LONG	NO	
timestamp	LONG	YES	

稳定币自动兑换划转 (TRADE)

响应

```
{
  "tranId": 118263407119,
  "status": "S"
}
```

`POST /sapi/v1/asset/convert-transfer`

稳定币和BUSD之间的自动划转

权重(UID): 5

参数

名称	类型	是否必需	描述
clientTranId	STRING	YES	用户自定义流水号，唯一标志，限制最短长度为20
asset	STRING	YES	当前资产
amount	BigDecimal	YES	数量必须为正数
targetAsset	String	YES	目标资产
accountType	String	NO	仅支持MAIN和CARD，如果为空，默认查询主账户MAIN

- 如果clientTranId你之前使用过，不会进行第二次自动转化，而是把之前划转的结果返回

稳定币自动兑换划转查询 (USER_DATA)

响应

```
{
  "total":3,
  "rows":
  [
    {
      "tranId":118263615991,
      "type":244,
      "time":1664442078000,
      "deductedAsset":"BUSD",
      "deductedAmount":"1",
      "targetAsset":"USDC",
      "targetAmount":"1",
      "status":"S",
      "accountType":"MAIN"
    },
    {
      "tranId":118263598801,
      "type":244,
      "time":1664442061000,
      "deductedAsset":"BUSD",
      "deductedAmount":"1",
      "targetAsset":"USDC",
      "targetAmount":"1",
      "status":"S",
      "accountType":"MAIN"
    },
    {
      "tranId":118263407119,
      "type":244,
      "time":1664441820000,
      "deductedAsset":"BUSD",
      "deductedAmount":"1"
    }
  ]
}
```

```

    "deductedAmount": "1",
    "targetAsset": "USDC",
    "targetAmount": "1",
    "status": "S",
    "accountType": "MAIN"
}
]
}

```

GET /sapi/v1/asset/convert-transfer/queryByPage

权重(UID): 5

参数

名称	类型	是否必需	描述
tranId	LONG	NO	流水号
clientTranId	STRING	NO	用户自定义流水号
asset	STRING	NO	不传或者空字符串查全部, 匹配扣除币种和目标币种
startTime	LONG	YES	开始时间 (包含) , 单位 : 毫秒
endTime	LONG	YES	结束时间 (不包含) , 单位 : 毫秒
accountType	STRING	NO	账户类型: MAIN-主账户。CARD-资金账户。如果传入则仅返回对应wallet的记录, 不传或者传null则返回该用户spot和card钱包的记录。
current	INTEGER	NO	当前页面, 默认1, 最小值为1
size	INTEGER	NO	页面大小, 默认10, 最大值为100

查询用户API Key权限 (USER_DATA)

响应

```
{
  "ipRestrict": false, // 是否限制ip访问
  "createTime": 1623840271000, // 创建时间
  "enableWithdrawals": false, // 此选项允许通过此api提现。开启提现选项必须添加IP访问限制过滤器
  "enableInternalTransfer": true, // 此选项授权此密钥在您的母账户和子账户之间划转资金
  "permitsUniversalTransfer": true, // 授权该密钥可用于专用的万向划转接口, 用以操作其支持的多种类型资金划转。各
  "enableVanillaOptions": false, // 欧式期权交易权限
  "enableReading": true,
  "enableFutures": false, // 合约交易权限, 需注意开通合约账户之前创建的API Key不支持合约API功能
  "enableMargin": false, // 此选项在全仓账户完成划转后可编辑
  "enableSpotAndMarginTrading": false, // 现货和杠杆交易权限
  "tradingAuthorityExpirationTime": 1628985600000 // 现货和杠杆交易权限到期时间, 如果没有则不返回该字段
}
```

GET /sapi/v1/account/apiRestrictions (HMAC SHA256)

权重(IP): 1**参数:**

名称	类型	是否必需	描述
recvWindow	LONG	NO	
timestamp	LONG	YES	

子母账户接口

- 这些接口的文档适用于企业账户。
- 关于如何成为企业用户，请参考: [企业账户申请](#)。

创建虚拟子账户 (适用主账户)

响应:

```
{
  "email": "addsdd_virtual@aasaixwqnoemail.com"
}
```

`POST /sapi/v1/sub-account/virtualSubAccount (HMAC SHA256)`

权重(IP): 1**参数:**

名称	类型	是否必需	描述
subAccountString	STRING	YES	请输入字符串，我们将为您创建一个虚拟邮箱进行注册
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 该请求会为您的母账户生成一个虚拟子账户
- 您需要为母账户apikey开通"允许现货及杠杆交易" 权限调用此接口

查询子账户列表 (适用主账户)

响应:

```
{
  "subAccounts": [
    {
      "email": "testsub@gmail.com",
      ...
    }
  ]
}
```

```

    "isFreeze":false,
    "createTime":1544433328000,
    "isManagedSubAccount": false,
    "isAssetManagementSubAccount": false
},
{
    "email":"virtual@oxebmvfonemail.com",
    "isFreeze":false,
    "createTime":1544433328000,
    "isManagedSubAccount": false,
    "isAssetManagementSubAccount": false
}
]
}

```

`GET /sapi/v1/sub-account/list (HMAC SHA256)`

权重(IP): 1

参数:

名称	类型	是否必需	描述
email	STRING	NO	Sub-account email
isFreeze	STRING	NO	true or false
page	INT	NO	默认: 1
limit	INT	NO	默认: 1, 最大: 200
recvWindow	LONG	NO	
timestamp	LONG	YES	

查询子账户现货资金划转历史 (适用主账户)

响应:

```

[
{
    "from":"aaa@test.com",
    "to":"bbb@test.com",
    "asset":"BTC",
    "qty":"10",
    "status": "SUCCESS",
    "tranId": 6489943656,
    "time":1544433328000
},
{
    "from":"bbb@test.com",
    "to":"ccc@test.com",
    "asset":"ETH",
    "qty":"2",
    "status": "SUCCESS",
}
]

```

```

        "tranId": 6489938713,
        "time":1544433328000
    }
]

```

GET /sapi/v1/sub-account/sub/transfer/history (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
fromEmail	STRING	NO	
toEmail	STRING	NO	
startTime	LONG	NO	
endTime	LONG	NO	
page	INT	NO	默认: 1
limit	INT	NO	默认: 500
recvWindow	LONG	NO	
timestamp	LONG	YES	

- fromEmail 和 toEmail 不可以同时发送
- 若 fromEmail 和 toEmail 都未传，默认返回fromEmail 为母账户的记录。

查询子账户合约资金划转历史 (适用主账户)

响应

```

{
    "success":true,
    "futuresType": 2,
    "transfers":[
        {
            "from":"aaa@test.com",
            "to":"bbb@test.com",
            "asset":"BTC",
            "qty":"1",
            "tranId":11897001102,
            "time":1544433328000
        },
        {
            "from":"bbb@test.com",
            "to":"ccc@test.com",
            "asset":"ETH",
            "qty":"2",
            "tranId":11631474902,
            "time":1544433328000
        }
    ]
}

```

```
[  
}]
```

`GET /sapi/v1/sub-account/futures/internalTransfer (HMAC SHA256)`

权重(IP): 1

参数:

名称	类型	是否必需	描述
email	STRING	YES	子账户邮箱 备注
futuresType	LONG	YES	1:USDT合约 , 2: 币本位合约
startTime	LONG	NO	默认返回100天内历史记录
endTime	LONG	NO	默认返回100天内历史记录
page	INT	NO	默认值: 1
limit	INT	NO	默认值: 50, 最大值 : 500
recvWindow	LONG	NO	
timestamp	LONG	YES	

执行子账户合约资金划转 (适用主账户)

响应

```
{
  "success":true,
  "txnid":"2934662589"
}
```

`POST /sapi/v1/sub-account/futures/internalTransfer (HMAC SHA256)`

权重(IP): 1

参数:

名称	类型	是否必需	描述
fromEmail	STRING	YES	发送者邮箱 备注
toEmail	STRING	YES	接收者邮箱 备注
futuresType	LONG	YES	1:USDT合约 , 2: 币本位合约
asset	STRING	YES	
amount	DECIMAL	YES	
recvWindow	LONG	NO	

名称	类型	是否必需	描述
timestamp	LONG	YES	

- 每个母账户每分钟上限2000次
- 您期货钱包中须有足够保证金余额才能执行转账

查询子账户资产 (适用主账户)

响应

```
{
  "balances": [
    {
      "asset": "ADA",
      "free": 10000,
      "locked": 0
    },
    {
      "asset": "BNB",
      "free": 10003,
      "locked": 0
    },
    {
      "asset": "BTC",
      "free": 11467.6399,
      "locked": 0
    },
    {
      "asset": "ETH",
      "free": 10004.995,
      "locked": 0
    },
    {
      "asset": "USDT",
      "free": 11652.14213,
      "locked": 0
    }
  ]
}
```

GET /sapi/v3/sub-account/assets (HMAC SHA256)

权重(UID): 60

参数:

名称	类型	是否必需	描述
email	STRING	YES	子账户邮箱 备注
recvWindow	LONG	NO	
timestamp	LONG	YES	

查询子账户现货资产汇总 (适用主账户)

响应:

```
{
  "totalCount": 2,
  "masterAccountTotalAsset": "0.23231201",
  "spotSubUserAssetBtcVoList": [
    {
      "email": "sub123@test.com",
      "totalAsset": "9999.00000000"
    },
    {
      "email": "test456@test.com",
      "totalAsset": "0.00000000"
    }
  ]
}
```

获取BTC计价的子账户现货资产汇总。

GET /sapi/v1/sub-account/spotSummary (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
email	STRING	NO	子账户邮箱
page	LONG	NO	分页，默认 1
size	LONG	NO	单页条目数，默认 10, 最大 20
recvWindow	LONG	NO	
timestamp	LONG	YES	

获取子账户充值地址 (适用主账户)

响应

```
{
  "address": "TDunhSa7jkTNuKrusUTU1MUHtqXoBPKETV",
  "coin": "USDT",
  "tag": "",
  "url": "https://tronscan.org/#/address/TDunhSa7jkTNuKrusUTU1MUHtqXoBPKETV"
}
```

GET /sapi/v1/capital/deposit/subAddress (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
email	STRING	YES	子账户邮箱 备注
coin	STRING	YES	
network	STRING	NO	
recvWindow	LONG	NO	
timestamp	LONG	YES	

获取子账户充值记录 (适用主账户)

响应

```
[
  {
    "id": "769800519366885376",
    "amount": "0.001",
    "coin": "BNB",
    "network": "BNB",
    "status": 0,
    "address": "bnb136ns61fw4zs5hg4n85vdthaad7hq5m4gtkgf23",
    "addressTag": "101764890",
    "txId": "98A3EA560C6B3336D348B6C83F0F95ECE4F1F5919E94BD006E5BF3BF264FACFC",
    "insertTime": 1661493146000,
    "transferType": 0,
    "confirmTimes": "1/1",
    "unlockConfirm": 0,
    "walletType": 0
  },
  {
    "id": "769754833590042625",
    "amount": "0.50000000",
    "coin": "IOTA",
    "network": "IOTA",
    "status": 1,
    "address": "SIZ9VLMHWATXKV99LH99CIGFJFUMLEHGWVZVNNZXRJJVWBPHYWPPBOSDORZ9EQSHCZAMPVAPGFYQAUUV9DROOXJL",
    "addressTag": "",
    "txId": "ESBFVQUTPIWQNJSXPXFNHNYHSQNTGKRVKPRABQWTAXCDWOAKDKYWPTVG9BGXNVNKTLEJGESAVXIKIZ9999",
    "insertTime": 1599620082000,
    "transferType": 0,
    "confirmTimes": "1/1",
    "unlockConfirm": 0,
    "walletType": 0
  }
]
```

GET /sapi/v1/capital/deposit/subHisrec (HMAC SHA256)

权重(IP): 1**参数:**

名称	类型	是否必需	描述
email	STRING	YES	子账户邮箱 备注
coin	STRING	NO	
status	INT	NO	0(0:pending,6: credited but cannot withdraw, 1:success)
startTime	LONG	NO	
endTime	LONG	NO	
limit	INT	NO	
offset	INT	NO	default:0
recvWindow	LONG	NO	
timestamp	LONG	YES	
txId	STRING	NO	

查询子账户Margin/Futures状态 (适用主账户)

响应

```
[
  {
    "email": "123@test.com",          // user email
    "isSubUserEnabled": true,        // true or false
    "isUserActive": true,            // true or false
    "insertTime": 1570791523523,    // sub account create time
    "isMarginEnabled": true,         // true or false for margin
    "isFutureEnabled": true,         // true or false for futures.
    "mobile": 1570791523523        // user mobile number
  }
]
```

GET /sapi/v1/sub-account/status (HMAC SHA256)

权重(IP): 10**参数:**

名称	类型	是否必需	描述
email	STRING	NO	子账户邮箱 备注
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 如果不提交子账户email，返回所有子账户情况。

为子账户开通Margin (适用主账户)

响应

```
{
  "email": "123@test.com",
  "isMarginEnabled": true
}
```

`POST /sapi/v1/sub-account/margin/enable (HMAC SHA256)`

权重(IP): 1

参数:

名称	类型	是否必需	描述
email	STRING	YES	子账户邮箱 备注
recvWindow	LONG	NO	
timestamp	LONG	YES	

查询子账户Margin账户详情 (适用主账户)

响应

```
{
  "email": "123@test.com",
  "marginLevel": "11.64405625",
  "totalAssetOfBtc": "6.82728457",
  "totalLiabilityOfBtc": "0.58633215",
  "totalNetAssetOfBtc": "6.24095242",
  "marginTradeCoeffVo":
  {
    "forceLiquidationBar": "1.10000000", // 强平风险率
    "marginCallBar": "1.50000000", // 补仓风险率
    "normalBar": "2.00000000" // 初始风险率
  },
  "marginUserAssetVoList": [
    {
      "asset": "BTC",
      "borrowed": "0.00000000",
      "free": "0.00499500",
      "interest": "0.00000000",
      "locked": "0.00000000",
      "netAsset": "0.00499500"
    }
  ]
}
```

```

    },
    {
        "asset": "BNB",
        "borrowed": "201.66666672",
        "free": "2346.50000000",
        "interest": "0.00000000",
        "locked": "0.00000000",
        "netAsset": "2144.83333328"
    },
    {
        "asset": "ETH",
        "borrowed": "0.00000000",
        "free": "0.00000000",
        "interest": "0.00000000",
        "locked": "0.00000000",
        "netAsset": "0.00000000"
    },
    {
        "asset": "USDT",
        "borrowed": "0.00000000",
        "free": "0.00000000",
        "interest": "0.00000000",
        "locked": "0.00000000",
        "netAsset": "0.00000000"
    }
]
}

```

`GET /sapi/v1/sub-account/margin/account (HMAC SHA256)`

权重(IP): 10

参数:

名称	类型	是否必需	描述
email	STRING	YES	子账户邮箱 备注
recvWindow	LONG	NO	
timestamp	LONG	YES	

查询子账户Margin账户汇总 (适用主账户)

响应

```
{
    "totalAssetOfBtc": "4.3333333",
    "totalLiabilityOfBtc": "2.11111112",
    "totalNetAssetOfBtc": "2.2222221",
    "subAccountList": [
        {
            "email": "123@test.com",
            "totalAssetOfBtc": "2.11111111",
            ...
        }
    ]
}
```

```

    "totalLiabilityOfBtc": "1.11111111",
    "totalNetAssetOfBtc": "1.00000000"
},
{
    "email": "345@test.com",
    "totalAssetOfBtc": "2.22222222",
    "totalLiabilityOfBtc": "1.00000001",
    "totalNetAssetOfBtc": "1.22222221"
}
]
}

```

`GET /sapi/v1/sub-account/margin/accountSummary (HMAC SHA256)`

权重(IP): 10

参数:

名称	类型	是否必需	描述
recvWindow	LONG	NO	
timestamp	LONG	YES	

为子账户开通Futures (适用主账户)

响应

```
{
    "email": "123@test.com",
    "isFuturesEnabled": true // true or false
}
```

`POST /sapi/v1/sub-account/futures/enable (HMAC SHA256)`

权重(IP): 1

参数:

名称	类型	是否必需	描述
email	STRING	YES	子账户邮箱 备注
recvWindow	LONG	NO	
timestamp	LONG	YES	

查询子账户Futures账户详情 (适用主账户)

响应

```
{
  "email": "abc@test.com",
  "asset": "USDT",
  "assets": [
    {
      "asset": "USDT",
      "initialMargin": "0.00000000",
      "maintenanceMargin": "0.00000000",
      "marginBalance": "0.88308000",
      "maxWithdrawAmount": "0.88308000",
      "openOrderInitialMargin": "0.00000000",
      "positionInitialMargin": "0.00000000",
      "unrealizedProfit": "0.00000000",
      "walletBalance": "0.88308000"
    }
  ],
  "canDeposit": true,
  "canTrade": true,
  "canWithdraw": true,
  "feeTier": 2,
  "maxWithdrawAmount": "0.88308000",
  "totalInitialMargin": "0.00000000",
  "totalMaintenanceMargin": "0.00000000",
  "totalMarginBalance": "0.88308000",
  "totalOpenOrderInitialMargin": "0.00000000",
  "totalPositionInitialMargin": "0.00000000",
  "totalUnrealizedProfit": "0.00000000",
  "totalWalletBalance": "0.88308000",
  "updateTime": 1576756674610
}
```

GET /sapi/v1/sub-account/futures/account (HMAC SHA256)

权重(IP): 10

参数:

名称	类型	是否必需	描述
email	STRING	YES	子账户邮箱 备注
recvWindow	LONG	NO	
timestamp	LONG	YES	

查询子账户Futures账户汇总 (适用主账户)

响应

```
{
    "totalInitialMargin": "9.83137400",
    "totalMaintenanceMargin": "0.41568700",
    "totalMarginBalance": "23.03235621",
    "totalOpenOrderInitialMargin": "9.00000000",
    "totalPositionInitialMargin": "0.83137400",
    "totalUnrealizedProfit": "0.03219710",
    "totalWalletBalance": "22.15879444",
    "asset": "USD", //USDT和BUSD资产汇总
    "subAccountList": [
        {
            "email": "123@test.com",
            "totalInitialMargin": "9.00000000",
            "totalMaintenanceMargin": "0.00000000",
            "totalMarginBalance": "22.12659734",
            "totalOpenOrderInitialMargin": "9.00000000",
            "totalPositionInitialMargin": "0.00000000",
            "totalUnrealizedProfit": "0.00000000",
            "totalWalletBalance": "22.12659734",
            "asset": "USD" //USDT和BUSD资产汇总
        },
        {
            "email": "345@test.com",
            "totalInitialMargin": "0.83137400",
            "totalMaintenanceMargin": "0.41568700",
            "totalMarginBalance": "0.90575887",
            "totalOpenOrderInitialMargin": "0.00000000",
            "totalPositionInitialMargin": "0.83137400",
            "totalUnrealizedProfit": "0.03219710",
            "totalWalletBalance": "0.87356177",
            "asset": "USD"
        }
    ]
}
```

`GET /sapi/v1/sub-account/futures/accountSummary (HMAC SHA256)`

权重(IP): 1

参数:

名称	类型	是否必需	描述
recvWindow	LONG	NO	
timestamp	LONG	YES	

查询子账户合约持仓信息 (仅适用主账户)

响应

```
[
    {
```

```

    "entryPrice": "9975.12000",
    "leverage": "50",           // current initial leverage
    "maxNotional": "1000000",   // notional value limit of current initial leverage
    "liquidationPrice": "7963.54",
    "markPrice": "9973.50770517",
    "positionAmount": "0.010",
    "symbol": "BTCUSDT",
    "unrealizedProfit": "-0.01612295"
}
]

```

GET /sapi/v1/sub-account/futures/positionRisk (HMAC SHA256)

权重(IP): 10

参数:

名称	类型	是否必需	描述
email	STRING	YES	子账户邮箱 备注
recvWindow	LONG	NO	
timestamp	LONG	YES	

子账户Futures划转 (仅适用主账户)

响应

```
{
    "txnid": "2966662589"
}
```

POST /sapi/v1/sub-account/futures/transfer (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
email	STRING	YES	子账户邮箱 备注
asset	STRING	YES	划转资产, e.g., USDT
amount	DECIMAL	YES	划转数量
type	INT	YES	1: 由子账户的现货账户划转至其USDT本位合约账户; 2: 由子账户的USDT本位合约账户划转至其现货账户 ; 3:由子账户现货账户划转至其COIN本位合约账户 ; 4: 由子账户COIN本位合约账户划转至其现货账户
recvWindow	LONG	NO	

名称	类型	是否必需	描述
timestamp	LONG	YES	

子账户Margin划转 (仅适用主账户)

响应

```
{
  "txnid": "2966662589"
}
```

POST /sapi/v1/sub-account/margin/transfer (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
email	STRING	YES	子账户邮箱 备注
asset	STRING	YES	划转资产, e.g., USDT
amount	DECIMAL	YES	划转数量
type	INT	YES	1: 由子账户的现货账户划转至其杠杆账户; 2: 由子账户的杠杆账户划转至其现货账户
recvWindow	LONG	NO	
timestamp	LONG	YES	

向共同主账户下的子账户主动划转 (仅适用于子账户)

响应

```
{
  "txnid": "2966662589"
}
```

POST /sapi/v1/sub-account/transfer/subToSub (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
----	----	------	----

名称	类型	是否必需	描述
toEmail	STRING	YES	接收者子邮箱地址 备注
asset	STRING	YES	
amount	DECIMAL	YES	
recvWindow	LONG	NO	
timestamp	LONG	YES	

向主账户主动划转 (仅适用于子账户)

响应

```
{
  "txnid": "2966662589"
}
```

POST /sapi/v1/sub-account/transfer/subToMaster (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
asset	STRING	YES	
amount	DECIMAL	YES	
recvWindow	LONG	NO	
timestamp	LONG	YES	

查询子账户划转历史 (仅适用于子账户)

响应

```
[
  {
    "counterParty": "master",
    "email": "master@test.com",
    "type": 1, // 1 for transfer in , 2 for transfer out
    "asset": "BTC",
    "qty": "1",
    "fromAccountType": "SPOT",
    "toAccountType": "SPOT",
    "status": "SUCCESS",
    "tranId": 11798835829,
    "time": 1641428800000
  }
]
```

```

    "time":1544433325000
},
{
  "counterParty": "subAccount",
  "email": "sub2@test.com",
  "type": 1,
  "asset": "ETH",
  "qty": "2",
  "fromAccountType": "SPOT",
  "toAccountType": "SPOT",
  "status": "SUCCESS",
  "tranId": 11798829519,
  "time": 1544433326000
}
]

```

GET /sapi/v1/sub-account/transfer/subUserHistory (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
asset	STRING	NO	如不提供，返回所有asset划转记录
type	INT	NO	1: transfer in, 2: transfer out; 如不提供，返回transfer out方向划转记录
startTime	LONG	NO	
endTime	LONG	NO	
limit	INT	NO	默认值: 500
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 如果 startTime 和 endTime 均未发送， 默认只返回最近30天数据

子母账户万能划转 (适用主账户)

响应

```
{
  "tranId": 11945860693,
  "clientTranId": "test"
}
```

POST /sapi/v1/sub-account/universalTransfer (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
fromEmail	STRING	NO	
toEmail	STRING	NO	
fromAccountType	STRING	YES	"SPOT","USDT_FUTURE","COIN_FUTURE","MARGIN"(Cross),"ISOLATED_MARGIN"
toAccountType	STRING	YES	"SPOT","USDT_FUTURE","COIN_FUTURE","MARGIN"(Cross),"ISOLATED_MARGIN"
clientTranId	STRING	NO	不可重复
symbol	STRING	NO	仅在ISOLATED_MARGIN类型下使用
asset	STRING	YES	
amount	DECIMAL	YES	
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 需要开启母账户apikey“允许子母账户划转”权限。
- 若 fromEmail 未传，默认从母账户转出。
- 若 toEmail 未传，默认转入母账户。
- 最少指定fromEmail和toEmail 其中之一。
- 该接口支持的划转操作有：
 - 现货账户划转到现货账户、U本位合约账户、币本位合约账户（无论母账户或子账户）
 - 现货账户、U本位合约账户、币本位合约账户划转到现货账户（无论母账户或子账户）
 - 母账户现货账户划转到子账户杠杆全仓账户、杠杆逐仓账户
 - 子账户杠杆全仓账户、杠杆逐仓账户划转到母账户现货账户

查询子母账户万能划转历史 (适用主账户)

响应

```
{
  "result": [
    {
      "tranId": 92275823339,
      "fromEmail": "abctest@gmail.com",
      "toEmail": "deftest@gmail.com",
      "asset": "BNB",
      "amount": "0.01",
      "createTimeStamp": 1640317374000,
      "fromAccountType": "USDT_FUTURE",
      "toAccountType": "SPOT",
      "status": "SUCCESS",
      "clientTranId": "test"
    }
  ],
  "totalCount": 1
}
```

GET /sapi/v1/sub-account/universalTransfer (HMAC SHA256)

权重(IP): 1**参数:**

名称	类型	是否必需	描述
fromEmail	STRING	NO	
toEmail	STRING	NO	
clientTranId	STRING	NO	
startTime	LONG	NO	
endTime	LONG	NO	
page	INT	NO	默认 1
limit	INT	NO	默认 500, 最大 500
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 本查询接口只可以单边查询，fromEmail 和 toEmail 不能同时传入。
- 若 fromEmail 和 toEmail 都未传，默认返回 fromEmail 为母账户的划转记录。
- 若 startTime 和 endTime 都未传，则只可查询最近30天的记录。
- 查询时间范围最大不得超过30天。

查询子账户Futures账户详情V2 (适用主账户)

响应

USDT Margined Futures :

```
{
  "futureAccountResp": {
    "email": "abc@test.com",
    "assets": [
      {
        "asset": "USDT",
        "initialMargin": "0.00000000",
        "maintenanceMargin": "0.00000000",
        "marginBalance": "0.88308000",
        "maxWithdrawAmount": "0.88308000",
        "openOrderInitialMargin": "0.00000000",
        "positionInitialMargin": "0.00000000",
        "unrealizedProfit": "0.00000000",
        "walletBalance": "0.88308000"
      }
    ],
    "canDeposit": true,
    "canTrade": true,
    "canWithdraw": true,
    "feeTier": 2,
    "maxWithdrawAmount": "0.88308000",
    "maxWithdrawAmountUsdt": "0.88308000"
  }
}
```

```

    "totalInitialMargin": "0.00000000",
    "totalMaintenanceMargin": "0.00000000",
    "totalMarginBalance": "0.88308000",
    "totalOpenOrderInitialMargin": "0.00000000",
    "totalPositionInitialMargin": "0.00000000",
    "totalUnrealizedProfit": "0.00000000",
    "totalWalletBalance": "0.88308000",
    "updateTime": 1576756674610
}

```

COIN Margined Futures :

```

{
    "deliveryAccountResp": {
        "email": "abc@test.com",
        "assets": [
            {
                "asset": "BTC",
                "initialMargin": "0.00000000",
                "maintenanceMargin": "0.00000000",
                "marginBalance": "0.88308000",
                "maxWithdrawAmount": "0.88308000",
                "openOrderInitialMargin": "0.00000000",
                "positionInitialMargin": "0.00000000",
                "unrealizedProfit": "0.00000000",
                "walletBalance": "0.88308000"
            }
        ],
        "canDeposit": true,
        "canTrade": true,
        "canWithdraw": true,
        "feeTier": 2,
        "updateTime": 1598959682001
    }
}

```

GET /sapi/v2/sub-account/futures/account (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
email	STRING	YES	子账户邮箱 备注
futuresType	INT	YES	1:USDT Margined Futures, 2:COIN Margined Futures
recvWindow	LONG	NO	
timestamp	LONG	YES	

查询子账户Futures账户汇总V2 (适用主账户)

响应

USDT Margined Futures :

```
{
  "futureAccountSummaryResp": {
    "totalInitialMargin": "9.83137400",
    "totalMaintenanceMargin": "0.41568700",
    "totalMarginBalance": "23.03235621",
    "totalOpenOrderInitialMargin": "9.00000000",
    "totalPositionInitialMargin": "0.83137400",
    "totalUnrealizedProfit": "0.03219710",
    "totalWalletBalance": "22.15879444",
    "asset": "USD", //USDT和BUSD资产汇总
    "subAccountList": [
      {
        "email": "123@test.com",
        "totalInitialMargin": "9.00000000",
        "totalMaintenanceMargin": "0.00000000",
        "totalMarginBalance": "22.12659734",
        "totalOpenOrderInitialMargin": "9.00000000",
        "totalPositionInitialMargin": "0.00000000",
        "totalUnrealizedProfit": "0.00000000",
        "totalWalletBalance": "22.12659734",
        "asset": "USD" //USDT和BUSD资产汇总
      },
      {
        "email": "345@test.com",
        "totalInitialMargin": "0.83137400",
        "totalMaintenanceMargin": "0.41568700",
        "totalMarginBalance": "0.90575887",
        "totalOpenOrderInitialMargin": "0.00000000",
        "totalPositionInitialMargin": "0.83137400",
        "totalUnrealizedProfit": "0.03219710",
        "totalWalletBalance": "0.87356177",
        "asset": "USD"
      }
    ]
  }
}
```

COIN Margined Futures :

```
{
  "deliveryAccountSummaryResp": {
    "totalMarginBalanceOfBTC": "25.03221121",
    "totalUnrealizedProfitOfBTC": "0.12233410",
    "totalWalletBalanceOfBTC": "22.15879444",
    "asset": "BTC",
    "subAccountList": [
      {
        "email": "123@test.com",
        "totalMarginBalance": "22.12659734",
        "totalUnrealizedProfit": "0.00000000",
        "asset": "BTC"
      }
    ]
  }
}
```

```

        "totalWalletBalance": "22.12659734",
        "asset": "BTC"
    },
    {
        "email": "345@test.com",
        "totalMarginBalance": "0.90575887",
        "totalUnrealizedProfit": "0.03219710",
        "totalWalletBalance": "0.87356177",
        "asset": "BTC"
    }
]
}
}

```

`GET /sapi/v2/sub-account/futures/accountSummary (HMAC SHA256)`

权重(IP): 10

参数:

名称	类型	是否必需	描述
futuresType	INT	YES	1:USDT Margined Futures, 2:COIN Margined Futures
page	INT	NO	default:1
limit	INT	NO	default:10, max:20
recvWindow	LONG	NO	
timestamp	LONG	YES	

查询子账户合约持仓信息V2 (仅适用主账户)

响应

USDT Margined Futures :

```
{
    "futurePositionRiskVos": [
        {
            "entryPrice": "9975.12000",
            "leverage": "50", // current initial leverage
            "maxNotional": "1000000", // notional value limit of current initial leverage
            "liquidationPrice": "7963.54",
            "markPrice": "9973.50770517",
            "positionAmount": "0.010",
            "symbol": "BTCUSDT",
            "unrealizedProfit": "-0.01612295"
        }
    ]
}
```

COIN Margined Futures :

```
{
  "deliveryPositionRiskVos": [
    {
      "entryPrice": "9975.12000",
      "markPrice": "9973.50770517",
      "leverage": "20",
      "isolated": "false",
      "isolatedWallet": "9973.50770517",
      "isolatedMargin": "0.00000000",
      "isAutoAddMargin": "false",
      "positionSide": "BOTH",
      "positionAmount": "1.230",
      "symbol": "BTCUSD_201225",
      "unrealizedProfit": "-0.01612295"
    }
  ]
}
```

GET /sapi/v2/sub-account/futures/positionRisk (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
email	STRING	YES	子账户邮箱 备注
futuresType	INT	YES	1:USDT Margined Futures, 2:COIN Margined Futures
recvWindow	LONG	NO	
timestamp	LONG	YES	

为子账户开通杠杆代币 (适用母账户)

响应

```
{
  "email": "123@test.com",
  "enableBlvt": true
}
```

POST /sapi/v1/sub-account/blvt/enable (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
email	STRING	YES	Sub-account email
enableBlvt	BOOLEAN	YES	Only true for now

名称	类型	是否必需	描述
recvWindow	LONG	NO	
timestamp	LONG	YES	

为子账户API Key开启/关闭IP白名单 (适用母账户)

响应:

```
{
  "ipRestrict": "true",
  "ipList": [
    "0.0.0.0", // 0.0.0.0 为初始化ip（无额外意义），您需要用 `POST /sapi/v1/sub-account/subAccountApi/ipRestriction` 来设置
    "69.210.67.14",
    "8.34.21.10",
    "thirdPartyName" //只当您有开启三方IP白名单时才返回，您需要用 `POST /sapi/v1/sub-account/subAccountApi/ipRestriction` 来设置
  ],
  "updateTime": 1636371437000,
  "apiKey": "k5V491dtn4tszj6W3hystegdfvmGbqDzjmKCTpTvC0G74WhK7yd4rfCTo4lShf"
}
```

POST /sapi/v1/sub-account/subAccountApi/ipRestriction (HMAC SHA256)

权重(UID): 3000

参数:

名称	类型	是否必需	描述
email	STRING	YES	Sub-account email
subAccountApiKey	STRING	YES	
ipRestrict	BOOLEAN	YES	true or false
recvWindow	LONG	NO	
timestamp	LONG	YES	

为子账户API Key添加IP白名单 (适用母账户)

响应:

```
{
  "ip": [
    "8.34.21.101,5.24.40.1",
    "thirdPartyName" //只当您有开启三方IP白名单且添加了三方IP白名单时才返回，您需要用 `POST /sapi/v1/sub-account/subAccountApi/ipRestriction` 来设置
  ],
  "updateTime": 1636369557189,
}
```

```

    "apiKey": "k5V49ldtn4tszj6W3hystegdfvmGbqDzjmKCTpTvC0G74WhK7yd4rfCTo4lShf"
}

```

POST /sapi/v1/sub-account/subAccountApi/ipRestriction/ipList (HMAC SHA256)

您需要确保已经使用这个接口 **POST /sapi/v1/sub-account/subAccountApi/ipRestriction** 开启了ip白名单功能，才可添加ip地址

权重(UID): 3000

参数:

名称	类型	是否必需	描述
email	STRING	YES	Sub-account email
subAccountApiKey	STRING	YES	
ipAddress	STRING	NO	可批量添加，用逗号分隔。单API Key最多可添加30个
recvWindow	LONG	NO	
timestamp	LONG	YES	

查询子账户API Key IP白名单 (适用母账户)

响应:

```

{
    "ipRestrict": "true",
    "ipList": [
        "69.210.67.14",
        "8.34.21.10"
    ],
    "updateTime": 1636371437000,
    "apiKey": "k5V49ldtn4tszj6W3hystegdfvmGbqDzjmKCTpTvC0G74WhK7yd4rfCTo4lShf"
}

```

GET /sapi/v1/sub-account/subAccountApi/ipRestriction (HMAC SHA256)

权重(UID): 3000

参数:

名称	类型	是否必需	描述
email	STRING	YES	Sub-account email
subAccountApiKey	STRING	YES	
recvWindow	LONG	NO	
timestamp	LONG	YES	

删除子账户API Key IP白名单 (适用母账户)

响应:

```
{
  "ipRestrict": "true",
  "ipList": [
    "69.210.67.14",
    "8.34.21.10",
    "thirdPartyName" // 只当您有开启三方IP白名单且添加了三方IP白名单时才返回，您需要用 `POST /sapi/v1/sub-account/ip-restriction/ip-list` 来查询三方IP白名单
  ],
  "updateTime": 1636371437000,
  "apiKey": "k5V49ldtn4tszj6W3hystegdfvmGbqDzjmkCtpTvC0G74WhK7yd4rfCTo4lShf"
}
```

`DELETE /sapi/v1/sub-account/subAccountApi/ipRestriction/ipList (HMAC SHA256)`

权重(UID): 3000

参数:

名称	类型	是否必需	描述
email	STRING	YES	Sub-account email
subAccountApiKey	STRING	YES	
ipAddress	STRING	NO	可批量删除，用逗号分隔
recvWindow	LONG	NO	
timestamp	LONG	YES	

投资人账户为托管子账户充值资产 (适用投资人母账户)

响应

```
{
  "tranId": 66157362489
}
```

`POST /sapi/v1/managed-subaccount/deposit (HMAC SHA256)`

权重(IP): 1

参数:

名称	类型	是否必需	描述
toEmail	STRING	YES	
asset	STRING	YES	

名称	类型	是否必需	描述
amount	DECIMAL	YES	
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 您需要开通API Key允许现货和杠杆交易权限

投资人账户查询托管子账户资产 (适用投资人母账户)

响应

```
[
  {
    "coin": "INJ", //币种
    "name": "Injective Protocol", //名称
    "totalBalance": "0", //总资产
    "availableBalance": "0", //可用资产
    "inOrder": "0", //下单冻结
    "btcValue": "0" //btc估值
  },
  {
    "coin": "FILDOWN",
    "name": "FILDOWN",
    "totalBalance": "0",
    "availableBalance": "0",
    "inOrder": "0",
    "btcValue": "0"
  }
]
```

GET /sapi/v1/managed-subaccount/asset (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
email	STRING	YES	
recvWindow	LONG	NO	
timestamp	LONG	YES	

投资人账户为托管子账户提币资产 (适用投资人母账户)

响应

```
{
    "tranId":66157362489
}
```

POST /sapi/v1/managed-subaccount/withdraw (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
fromEmail	STRING	YES	
asset	STRING	YES	
amount	DECIMAL	YES	
transferDate	LONG	NO	提币会自动发生在选择的日期(UTC0) , 如果没有选择日期 , 提币会立即生效
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 您需要开通API Key 允许现货和杠杆交易权限

查询托管子账户资产快照 (适用投资人母账户)

响应

```
{
    "code":200, // 200表示返回正确, 否则即为错误码
    "msg":"",
    "snapshotVos":[
        {
            "data":{
                "balances":[
                    {
                        "asset":"BTC",
                        "free":"0.09905021",
                        "locked":"0.00000000"
                    },
                    {
                        "asset":"USDT",
                        "free":"1.89109409",
                        "locked":"0.00000000"
                    }
                ],
                "totalAssetOfBtc":"0.09942700"
            },
            "type":"spot",
            "updateTime":1576281599000
        }
    ]
}
```

}

或

```
{
  "code":200, // 200表示返回正确，否则即为错误码
  "msg":"",
  "snapshotVos":[
    {
      "data":{
        "marginLevel":"2748.02909813",
        "totalAssetOfBtc":"0.00274803",
        "totalLiabilityOfBtc":"0.00000100",
        "totalNetAssetOfBtc":"0.00274750",
        "userAssets":[
          {
            "asset":"XRP",
            "borrowed":"0.00000000",
            "free":"1.00000000",
            "interest":"0.00000000",
            "locked":"0.00000000",
            "netAsset":"1.00000000"
          }
        ]
      },
      "type":"margin",
      "updateTime":1576281599000
    }
  ]
}
```

或

```
{
  "code":200, // 200表示返回正确，否则即为错误码
  "msg":"",
  "snapshotVos":[
    {
      "data":{
        "assets":[
          {
            "asset":"USDT",
            "marginBalance":"118.99782335",
            "walletBalance":"120.23811389"
          }
        ],
        "position":[
          {
            "entryPrice":"7130.41000000",
            "markPrice":"7257.66239673",
            "positionAmt":"0.01000000",
            "symbol":"BTCUSDT",
            "unRealizedProfit":"1.24029054" //只显示开仓当时的未实现盈亏，不会实时更新，可以忽略
          }
        ]
      }
    }
  ]
}
```

```

    },
    "type": "futures",
    "updateTime": 1576281599000
}
]
}

```

`GET /sapi/v1/managed-subaccount/accountSnapshot (HMAC SHA256)`

权重(IP): 2400

参数:

名称	类型	是否必需	描述
email	STRING	YES	
type	STRING	YES	"SPOT" (现货), "MARGIN" (全仓), "FUTURES" (U本位合约)
startTime	LONG	NO	
endTime	LONG	NO	
limit	INT	NO	min 7, max 30, default 7
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 查询时间范围最大不得超过30天
- 仅支持查询最近 1 个月数据
- 若 startTime 和 endTime 没传，则默认返回最近7天数据

行情接口

测试服务器连通性

响应

```
{}
```

`GET /api/v3/ping`

测试能否联通 Rest API。

权重(IP): 1

参数:

NONE

数据源: 缓存

获取服务器时间

响应

```
{  
    "serverTime": 1499827319559  
}
```

GET /api/v3/time

测试能否联通 Rest API 并 获取服务器时间。

权重(IP): 1

参数:

NONE

数据源: 缓存

交易规范信息

响应

```
{  
    "timezone": "UTC",  
    "serverTime": 1565246363776,  
    "rateLimits": [  
        {  
            //这些在"限制种类 (rateLimitType)"下的"枚举定义"部分中定义  
            //所有限制都是可选的  
        }  
    ],  
    "exchangeFilters": [  
        //这些是"过滤器"部分中定义的过滤器  
        //所有限制都是可选的  
    ],  
    "symbols": [  
        {  
            "symbol": "ETHBTC",  
            "status": "TRADING",  
            "baseAsset": "ETH",  
            "baseAssetPrecision": 8,  
            "quoteAsset": "BTC",  
            "quotePrecision": 8,  
            "quoteAssetPrecision": 8,  
            "orderTypes": [  
                "LIMIT",  
                "LIMIT_MAKER",  
                "MARKET",  
                "STOP_LOSS",  
                "STOP_LOSS_LIMIT",  
                "TAKE_PROFIT",  
            ]  
        }  
    ]  
}
```

```

        "TAKE_PROFIT_LIMIT"
    ],
    "icebergAllowed": true,
    "ocoAllowed": true,
    "quoteOrderQtyMarketAllowed": false,
    "allowTrailingStop": false,
    "isSpotTradingAllowed": true,
    "isMarginTradingAllowed": true,
    "cancelReplaceAllowed": false,
    "filters": [
        //这些在"过滤器"部分中定义
        //所有限制都是可选的
    ],
    "permissions": [
        "SPOT",
        "MARGIN"
    ]
}
]
}
}

```

`GET /api/v3/exchangeInfo`

获取交易规则和交易对信息。

权重(IP): 10

参数:

有四种用法

用法	举例
不需要交易对	<code>curl -X GET "https://api.binance.com/api/v3/exchangeInfo"</code>
单个交易对	<code>curl -X GET "https://api.binance.com/api/v3/exchangeInfo?symbol=BNBBTC"</code>
多个交易对	<code>curl -X GET "https://api.binance.com/api/v3/exchangeInfo?symbols=%5B%22BNBBTC%22,%22BTCUSDT%22%5D"</code> 或者 <code>curl -g -X GET 'https://api.binance.com/api/v3/exchangeInfo?symbols=[\"BTCUSDT\",\"BNBBTC\"]'</code>
交易权限	<code>curl -X GET "https://api.binance.com/api/v3/exchangeInfo?permissions=SPOT"</code> 或者 <code>curl -X GET "https://api.binance.com/api/v3/exchangeInfo?permissions=%5B%22MARGIN%22%2C%22LEVERAGED%22%5D"</code> 或者 <code>curl -g -X GET 'https://api.binance.com/api/v3/exchangeInfo?permissions=[\"MARGIN\",\"LEVERAGED\"]'</code>

备注:

- 如果参数 `symbol` 或者 `symbols` 提供的交易对不存在, 系统会返回错误并提示交易对不正确.
- 所有的参数都是可选的.
- `permissions` 支持单个或者多个值, 比如 `SPOT`, `["MARGIN", "LEVERAGED"]`.
- 如果 `permissions` 值没有提供, 其默认值为 `["SPOT", "MARGIN", "LEVERAGED"]`.
 - 如果想取接口 `GET /api/v3/exchangeInfo` 的所有交易对, 则需要设置此参数的所有可能交易权限值, 比如 `permissions=[\"SPOT\", \"MARGIN\", \"LEVERAGED\", \"TRD_GRP_002\", \"TRD_GRP_003\", \"TRD_GRP_004\", \"TRD_GRP_005\"]`

数据源: 缓存

深度信息

响应

```
{
  "lastUpdateId": 1027024,
  "bids": [
    [
      "4.00000000",      // 价位
      "431.00000000"    // 挂单量
    ]
  ],
  "asks": [
    [
      "4.00000200",
      "12.00000000"
    ]
  ]
}
```

`GET /api/v3/depth`

权重(IP):

基于限制调整:

限制	权重
1-100	1
101-500	5
501-1000	10
1001-5000	50

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
limit	INT	NO	默认 100; 最大 5000. 可选值:[5, 10, 20, 50, 100, 500, 1000, 5000] 如果 limit > 5000, 最多返回5000条数据.

数据源: 缓存

近期成交列表

响应

```
[
  {
```

```

    "id": 28457,
    "price": "4.00000100",
    "qty": "12.00000000",
    "time": 1499865549590, // 交易成交时间，和websocket中的T一致。
    "isBuyerMaker": true,
    "isBestMatch": true
}
]

```

GET /api/v3/trades

获取近期成交

权重(IP): 1

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
limit	INT	NO	默认 500; 最大值 1000.

数据源: 缓存

查询历史成交 (MARKET_DATA)

响应

```

[
{
    "id": 28457,
    "price": "4.00000100",
    "qty": "12.00000000",
    "quoteQty": "48.000012",
    "time": 1499865549590,
    "isBuyerMaker": true,
    "isBestMatch": true
}
]

```

GET /api/v3/historicalTrades

获取历史成交。

权重(IP): 5

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
limit	INT	NO	默认 500; 最大值 1000.

名称	类型	是否必需	描述
fromId	LONG	NO	从哪一条成交id开始返回. 缺省返回最近的成交记录。

数据源: 数据库

近期成交(归集)

响应

```
[
  [
    {
      "a": 26129,          // 归集成交ID
      "p": "0.01633102",  // 成交价
      "q": "4.70443515", // 成交量
      "f": 27781,         // 被归集的首个成交ID
      "l": 27781,         // 被归集的末个成交ID
      "T": 1498793709153, // 成交时间
      "m": true,          // 是否为主动卖出单
      "M": true           // 是否为最优撮合单(可忽略, 目前总为最优撮合)
    }
  ]
]
```

GET /api/v3/aggTrades

归集交易与逐笔交易的区别在于，同一价格、同一方向、同一时间的trade会被聚合为一条

权重(IP): 1

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
fromId	LONG	NO	从包含fromId的成交id开始返回结果
startTime	LONG	NO	从该时刻之后的成交记录开始返回结果
endTime	LONG	NO	返回该时刻为止的成交记录
limit	INT	NO	默认 500; 最大 1000.

- 如果发送startTime和endTime，间隔必须小于一小时。
- 如果没有发送任何筛选参数(fromId, startTime,endTime)，默认返回最近的成交记录
- 如果一个trade有下面的值，表示这是一个重复的记录，并被标记为无效(invalid):
 - p = '0' // price
 - q = '0' // qty
 - f = -1 // first_trade_id
 - l = -1 // last_trade_id

数据源: 数据库

K线数据

响应

```
[
  [
    1499040000000,           // k线开盘时间
    "0.01634790",            // 开盘价
    "0.80000000",            // 最高价
    "0.01575800",            // 最低价
    "0.01577100",            // 收盘价(当前k线未结束的即为最新价)
    "148976.11427815",       // 成交量
    1499644799999,           // k线收盘时间
    "2434.19055334",         // 成交额
    308,                      // 成交笔数
    "1756.87402397",         // 主动买入成交量
    "28.46694368",           // 主动买入成交额
    "17928899.62484339"     // 请忽略该参数
  ]
]
```

`GET /api/v3/klines`

每根K线代表一个交易对。

每根K线的开盘时间可视为唯一ID

权重(IP): 1

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
interval	ENUM	YES	详见枚举定义 : K线间隔
startTime	LONG	NO	
endTime	LONG	NO	
limit	INT	NO	默认 500; 最大 1000.

- 如果未发送 startTime 和 endTime , 默认返回最近的交易。

数据源: 数据库

当前平均价格

响应

```
{
  "mins": 5,
  "price": "9.35751834"
}
```

`GET /api/v3/avgPrice`

权重(IP): 1

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	

数据源: 缓存

UIK线数据

响应

```
[
  [
    1499040000000,           // k线开盘时间
    "0.01634790",            // 开盘价
    "0.80000000",            // 最高价
    "0.01575800",            // 最低价
    "0.01577100",            // 收盘价(当前K线未结束的即为最新价)
    "148976.11427815",       // 成交量
    1499644799999,           // k线收盘时间
    "2434.19055334",         // 成交额
    308,                      // 成交笔数
    "1756.87402397",         // 主动买入成交量
    "28.46694368",           // 主动买入成交额
    "0" // 请忽略该参数
  ]
]
```

`GET /api/v3/uiKlines`

请求参数与响应和k线接口相同。

`uiKlines` 返回修改后的k线数据，针对k线图的呈现进行了优化。

权重(IP): 1

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
interval	ENUM	YES	
startTime	LONG	NO	
endTime	LONG	NO	
limit	INT	NO	默认 500; 最大 1000.

- 如果未发送 startTime 和 endTime，默认返回最近的交易。

数据源: 数据库

24hr 价格变动情况

响应 - FULL

```
{
  "symbol": "BNBBTC",
  "priceChange": "-94.99999800",
  "priceChangePercent": "-95.960",
  "weightedAvgPrice": "0.29628482",
  "prevClosePrice": "0.10002000",
  "lastPrice": "4.00000200",
  "lastQty": "200.00000000",
  "bidPrice": "4.00000000",
  "bidQty": "100.00000000",
  "askPrice": "4.00000200",
  "askQty": "100.00000000",
  "openPrice": "99.00000000",
  "highPrice": "100.00000000",
  "lowPrice": "0.10000000",
  "volume": "8913.30000000",
  "quoteVolume": "15.30000000",
  "openTime": 1499783499040,
  "closeTime": 1499869899040,
  "firstId": 28385,    // 首笔成交id
  "lastId": 28460,    // 末笔成交id
  "count": 76         // 成交笔数
}
```

OR

```
[
  {
    "symbol": "BNBBTC",
    "priceChange": "-94.99999800",
    "priceChangePercent": "-95.960",
    "weightedAvgPrice": "0.29628482",
    "prevClosePrice": "0.10002000",
    "lastPrice": "4.00000200",
    "lastQty": "200.00000000",
    "bidPrice": "4.00000000",
    "bidQty": "100.00000000",
    "askPrice": "4.00000200",
    "askQty": "100.00000000",
    "openPrice": "99.00000000",
    "highPrice": "100.00000000",
    "lowPrice": "0.10000000",
    "volume": "8913.30000000",
    "quoteVolume": "15.30000000",
    "openTime": 1499783499040,
    "closeTime": 1499869899040,
```

```

    "firstId": 28385,
    "lastId": 28460,
    "count": 76
}
]

```

Response - MINI

```
{
  "symbol": "BNBBTC",           // 交易对
  "openPrice": "99.00000000",   // 间隔开盘价
  "highPrice": "100.00000000",  // 间隔最高价
  "lowPrice": "0.10000000",    // 间隔最低价
  "lastPrice": "4.00000200",   // 间隔收盘价
  "volume": "8913.30000000",   // 总交易量 (base asset)
  "quoteVolume": "15.30000000", // 总交易量 (quote asset)
  "openTime": 1499783499040,   // ticker间隔的开始时间
  "closeTime": 1499869899040,  // ticker间隔的结束时间
  "firstId": 28385,            // 统计时间内的第一笔trade id
  "lastId": 28460,             // 统计时间内的最后一笔trade id
  "count": 76                 // 统计时间内交易笔数
}
```

OR

```
[
{
  "symbol": "BNBBTC",
  "openPrice": "99.00000000",
  "highPrice": "100.00000000",
  "lowPrice": "0.10000000",
  "lastPrice": "4.00000200",
  "volume": "8913.30000000",
  "quoteVolume": "15.30000000",
  "openTime": 1499783499040,
  "closeTime": 1499869899040,
  "firstId": 28385,
  "lastId": 28460,
  "count": 76
},
{
  "symbol": "LTCBTC",
  "openPrice": "0.07000000",
  "highPrice": "0.07000000",
  "lowPrice": "0.07000000",
  "lastPrice": "0.07000000",
  "volume": "11.00000000",
  "quoteVolume": "0.77000000",
  "openTime": 1656908192899,
  "closeTime": 1656994592899,
  "firstId": 0,
  "lastId": 10,
  "count": 11
}
]
```

GET /api/v3/ticker/24hr

24 小时滚动窗口价格变动数据。请注意，不携带symbol参数会返回全部交易对数据，不仅数据庞大，而且权重极高

权重(IP):

参数	提供Symbol数量	权重
symbol	1	1
	不提供symbol	40
symbols	1-20	1
	21-100	20
	>= 101	40
	不提供symbol	40

参数:

名称	类型	是否强制要求	详情
symbol	STRING	NO	参数 'symbol' 和 'symbols' 不可以一起使用 如果不提供, 所有symbol的ticker数据都会返回.
symbols	STRING	NO	symbols参数可接受的格式 : ["BTCUSDT","BNBUSDT"] 或 %5B%22BTCUSDT%22,%22BNBUSDT%22%5D
type	ENUM	NO	可接受的参数: FULL or MINI. 如果不提供, 默认值为 FULL

数据源: 缓存

最新价格

响应

```
{
  "symbol": "LTCBTC",
  "price": "4.00000200"
}
```

OR

```
[
  {
    "symbol": "LTCBTC",
    "price": "4.00000200"
  },
  {
    "symbol": "ETHBTC",
    "price": "0.07946600"
  }
]
```

```

    }
]
```

`GET /api/v3/ticker/price`

获取交易对最新价格

权重(IP):

参数	Symbols数量	权重
symbol	1	1
	不提供symbol	2
symbols	不限	2

参数:

参数名	类型	是否强制	详情
symbol	STRING	NO	参数 `symbol` 和 `symbols` 不可以一起使用 如果都不提供, 所有symbol的价格数据都会返回.
symbols	STRING	NO	symbols参数可接受的格式 : ["BTCUSDT","BNBUSDT"] 或 %5B%22BTCUSDT%22,%22BNBUSDT%22%5D

- 不发送交易对参数，则会返回所有交易对信息

数据源: 缓存

当前最优挂单

响应

```
{
  "symbol": "LTCBTC",
  "bidPrice": "4.00000000",
  "bidQty": "431.00000000",
  "askPrice": "4.0000200",
  "askQty": "9.00000000"
}
```

OR

```
[
  {
    "symbol": "LTCBTC",
    "bidPrice": "4.00000000",
    "bidQty": "431.00000000",
    "askPrice": "4.0000200",
    "askQty": "9.00000000"
  },
  {
    "symbol": "LTCBTC",
    "bidPrice": "4.00000000",
    "bidQty": "431.00000000",
    "askPrice": "4.0000200",
    "askQty": "9.00000000"
  }
]
```

```

    "symbol": "ETHBTC",
    "bidPrice": "0.07946700",
    "bidQty": "9.00000000",
    "askPrice": "100000.00000000",
    "askQty": "1000.00000000"
}
]

```

`GET /api/v3/ticker/bookTicker`

返回当前最优的挂单(最高买单，最低卖单)

权重(IP):

参数	Symbols数量	权重
symbol	1	1
	不提供symbol	2
symbols	不限	2

参数:

参数名	类型	是否强制	详情
symbol	STRING	NO	参数 `symbol` 和 `symbols` 不可以一起使用 如果都不提供, 所有symbol的价格数据都会返回.
symbols	STRING	NO	symbols参数可接受的格式 : ["BTCUSDT","BNBUSDT"] 或 %5B%22BTCUSDT%22,%22BNBUSDT%22%5D

数据源: 缓存

滚动窗口价格变动统计

响应 - FULL

```
{
  "symbol": "BNBBTC",
  "priceChange": "-8.00000000", // 价格变化
  "priceChangePercent": "-88.889", // 价格变化百分比
  "weightedAvgPrice": "2.60427807",
  "openPrice": "9.00000000",
  "highPrice": "9.00000000",
  "lowPrice": "1.00000000",
  "lastPrice": "1.00000000",
  "volume": "187.00000000",
  "quoteVolume": "487.00000000",
  "openTime": 1641859200000, // ticker的开始时间
  "closeTime": 1642031999999, // ticker的结束时间
  "firstId": 0, // 统计时间内的第一笔trade id
  "lastId": 60,
  "count": 61 // 统计时间内交易笔数
}
```

}

或者

```
[
  {
    "symbol": "BTCUSDT",
    "priceChange": "-154.13000000",
    "priceChangePercent": "-0.740",
    "weightedAvgPrice": "20677.46305250",
    "openPrice": "20825.27000000",
    "highPrice": "20972.46000000",
    "lowPrice": "20327.92000000",
    "lastPrice": "20671.14000000",
    "volume": "72.65112300",
    "quoteVolume": "1502240.91155513",
    "openTime": 1655432400000,
    "closeTime": 1655446835460,
    "firstId": 11147809,
    "lastId": 11149775,
    "count": 1967
  },
  {
    "symbol": "BNBBTC",
    "priceChange": "0.00008530",
    "priceChangePercent": "0.823",
    "weightedAvgPrice": "0.01043129",
    "openPrice": "0.01036170",
    "highPrice": "0.01049850",
    "lowPrice": "0.01033870",
    "lastPrice": "0.01044700",
    "volume": "166.67000000",
    "quoteVolume": "1.73858301",
    "openTime": 1655432400000,
    "closeTime": 1655446835460,
    "firstId": 2351674,
    "lastId": 2352034,
    "count": 361
  }
]
```

响应 - MINI

```
{
  "symbol": "LTCBTC",
  "openPrice": "0.10000000",
  "highPrice": "2.00000000",
  "lowPrice": "0.10000000",
  "lastPrice": "2.00000000",
  "volume": "39.00000000",
  "quoteVolume": "13.40000000", // 此k线内所有交易的price(价格) x volume(交易量)的总和
  "openTime": 1656986580000, // ticker窗口的开始时间
  "closeTime": 1657001016795, // ticker窗口的结束时间
  "firstId": 0, // 首笔成交id
  "lastId": 34,
```

```
"count": 35 // 统计时间内交易笔数
}
```

OR

```
[
{
  "symbol": "BNBBTC",
  "openPrice": "0.10000000",
  "highPrice": "2.00000000",
  "lowPrice": "0.10000000",
  "lastPrice": "2.00000000",
  "volume": "39.00000000",
  "quoteVolume": "13.40000000", // 此k线内所有交易的price(价格) x volume(交易量)的总和
  "openTime": 1656986880000, // ticker窗口的开始时间
  "closeTime": 1657001297799, // ticker窗口的结束时间
  "firstId": 0, // 首笔成交id
  "lastId": 34,
  "count": 35 // 统计时间内交易笔数
},
{
  "symbol": "LTCBTC",
  "openPrice": "0.07000000",
  "highPrice": "0.07000000",
  "lowPrice": "0.07000000",
  "lastPrice": "0.07000000",
  "volume": "33.00000000",
  "quoteVolume": "2.31000000",
  "openTime": 1656986880000,
  "closeTime": 1657001297799,
  "firstId": 0,
  "lastId": 32,
  "count": 33
}
]
```

`GET /api/v3/ticker`

注意: 此接口和 `GET /api/v3/ticker/24hr` 有所不同.

此接口统计的时间范围比请求的 `windowSize` 多不超过 59999ms.

接口的 `openTime` 是某一分钟的起始，而结束是当前的时间. 所以实际的统计区间会比请求的时间窗口多不超过 59999ms.

比如, 结束时间 `closeTime` 是 1641287867099 (January 04, 2022 09:17:47:099 UTC), `windowSize` 为 `1d`. 那么开始时间 `openTime` 则为 1641201420000 (January 3, 2022, 09:17:00 UTC)

权重(IP): 2/交易对.

如果 `symbols` 请求的交易对超过 50, 上限是 100.

参数

Name	Type	Mandatory	Description
symbol	STRING	YES	提供 symbol 或者 symbols 其中之一 symbols 可以传入的格式: ["BTCUSDT", "BNBUSDT"]
symbols			

or
%5B%22BTCUSDT%22,%22BNBUSDT%22%5D

symbols 允许最多100个交易对

默认为 1d

windowSize 支持的值:

如果是分钟: 1m, 2m....59m

如果是小时: 1h, 2h....23h

如果是天: 1d...7d

不可以组合使用, 比如1d2h

windowSize ENUM NO

type ENUM NO 可接受的参数: FULL or MINI.

如果不提供, 默认值为 FULL

数据源: 数据库

Websocket 行情推送

- 本篇所列出的所有wss接口的baseUrl为: **wss://stream.binance.com:9443** 或者 **wss://stream.binance.com:443**
- Streams有单一原始 stream 或组合 stream
- 单一原始 streams 格式为 **/ws/<streamName>**
- 组合streams的URL格式为 **/stream?streams= <streamName1>/<streamName2>/<streamName3>**
- 订阅组合streams时, 事件payload会以这样的格式封装: **{"stream": "<streamName>", "data": <rawPayload>}**
- stream名称中所有交易对均为 **小写**
- 每个到 **stream.binance.com** 的链接有效期不超过24小时, 请妥善处理断线重连。
- 每3分钟, 服务端会发送ping帧, 客户端应当在10分钟内回复pong帧, 否则服务端会主动断开链接。允许客户端发送不成对的pong帧(即客户端可以以高于10分钟每次的频率发送pong帧保持链接)。

实时订阅/取消数据流

- 以下数据可以通过websocket发送以实现订阅或取消订阅数据流。示例如下。
- 响应内容中的 **id** 是无符号整数, 作为往来信息的唯一标识。
- 如果相应内容中的 **result** 为 **null**, 表示请求发送成功。

订阅一个信息流

响应

```
{
  "result": null,
  "id": 1
}
```

• 请求

```
{
  "method": "SUBSCRIBE",
  "params": [
    "btcusdt@aggTrade",
```

```
"btcusdt@depth"
],
"id": 1
}
```

取消订阅一个信息流

响应

```
{
  "result": null,
  "id": 312
}
```

- **请求**

```
{
  "method": "UNSUBSCRIBE",
  "params": [
    "btcusdt@depth"
  ],
  "id": 312
}
```

已订阅信息流

响应

```
{
  "result": [
    "btcusdt@aggTrade"
  ],
  "id": 3
}
```

- **请求**

```
{
  "method": "LIST_SUBSCRIPTIONS",
  "id": 3
}
```

设定属性

当前，唯一可以设置的属性是设置是否启用 `combined` ("组合") 信息流。

当使用 `/ws/` ("原始信息流") 进行连接时，`combined` 属性设置为 `false`，而使用 `/stream/` 进行连接时则将属性设置为 `true`。

响应

```
{
  "result": null,
  "id": 5
}
```

- **请求**

```
{
  "method": "SET_PROPERTY",
  "https://binance-docs.github.io/apidocs/spot/cn/#185368440e
```

```
"params":  
[  
  "combined",  
  true  
,  
  "id": 5  
]
```

检索属性

响应

```
{  
  "result": true, // Indicates that combined is set to true.  
  "id": 2  
}
```

- **请求**

```
{
  "method": "GET_PROPERTY",
  "params":  
[  
  "combined"
],  
  "id": 2
}
```

错误信息

错误信息	描述
{"code": 0, "msg": "Unknown property"}	SET_PROPERTY 或 GET_PROPERTY 中应用的参数无效
{"code": 1, "msg": "Invalid value type: expected Boolean", "id": "%s'}	仅接受 true 或 false
{"code": 2, "msg": "Invalid request: property name must be a string"}	提供的属性名无效
{"code": 2, "msg": "Invalid request: request ID must be an unsigned integer"}	参数 id 未提供或 id 值是无效类型
{"code": 2, "msg": "Invalid request: unknown variant %s, expected one of SUBSCRIBE, UNSUBSCRIBE, LIST_SUBSCRIPTIONS, SET_PROPERTY, GET_PROPERTY at line 1 column 28"}	错字提醒，或提供的值不是预期类型
{"code": 2, "msg": "Invalid request: too many parameters"}	数据中提供了不必要的参数
{"code": 2, "msg": "Invalid request: property name must be a string"}	未提供属性名
{"code": 2, "msg": "Invalid request: missing field method at line 1 column 73"}	数据未提供 method
{"code": 3, "msg": "Invalid JSON: expected value at line %s column %s"}	JSON 语法有误。

归集交易流

Payload:

```
{
  "e": "aggTrade", // 事件类型
  "E": 123456789, // 事件时间
  "s": "BNBBTC", // 交易对
  "a": 12345, // 归集交易ID
  "p": "0.001", // 成交价格
  "q": "100", // 成交数量
  "f": 100, // 被归集的首个交易ID
  "l": 105, // 被归集的末次交易ID
  "T": 123456785, // 成交时间
  "m": true, // 买方是否是做市方。如true，则此次成交是一个主动卖出单，否则是一个主动买单。
  "M": true // 请忽略该字段
}
```

归集交易 stream 推送交易信息，是对单一订单的集合。

Stream 名称: <symbol>@aggTrade

Update Speed: 实时

逐笔交易

Payload:

```
{
  "e": "trade", // 事件类型
  "E": 123456789, // 事件时间
  "s": "BNBBTC", // 交易对
  "t": 12345, // 交易ID
  "p": "0.001", // 成交价格
  "q": "100", // 成交数量
  "b": 88, // 买方的订单ID
  "a": 50, // 卖方的订单ID
  "T": 123456785, // 成交时间
  "m": true, // 买方是否是做市方。如true，则此次成交是一个主动卖出单，否则是一个主动买单。
  "M": true // 请忽略该字段
}
```

Stream Name: <symbol>@trade

逐笔交易推送每一笔成交的信息。**成交**，或者说交易的定义是仅有一个吃单者与一个挂单者相互交易

K线 Streams

Payload:

```
{
  "e": "kline", // 事件类型
  "E": 123456789, // 事件时间
  "s": "BNBBTC", // 交易对
  "k": {
```

```

    "t": 123400000, // 这根K线的起始时间
    "T": 123460000, // 这根K线的结束时间
    "s": "BNBBTC", // 交易对
    "i": "1m", // K线间隔
    "f": 100, // 这根K线期间第一笔成交ID
    "L": 200, // 这根K线期间末一笔成交ID
    "o": "0.0010", // 这根K线期间第一笔成交价
    "c": "0.0020", // 这根K线期间末一笔成交价
    "h": "0.0025", // 这根K线期间最高成交价
    "l": "0.0015", // 这根K线期间最低成交价
    "v": "1000", // 这根K线期间成交量
    "n": 100, // 这根K线期间成交笔数
    "x": false, // 这根K线是否完结(是否已经开始下一根K线)
    "q": "1.0000", // 这根K线期间成交额
    "V": "500", // 主动买入的成交量
    "Q": "0.500", // 主动买入的成交额
    "B": "123456" // 忽略此参数
}
}

```

K线stream逐秒推送所请求的K线种类(最新一根K线)的更新。

Stream Name: <symbol>@kline_<interval>

Update Speed: 2000ms

K线图间隔参数:

s -> 秒; m -> 分钟; h -> 小时; d -> 天; w -> 周; M -> 月

- 1s
- 1m
- 3m
- 5m
- 15m
- 30m
- 1h
- 2h
- 4h
- 6h
- 8h
- 12h
- 1d
- 3d
- 1w
- 1M

按 Symbol 的精简Ticker

Payload:

```
{
    "e": "24hrMiniTicker", // 事件类型
    "E": 123456789, // 事件时间
    "s": "BNBBTC", // 交易对
    "c": "0.0025", // 最新成交价格
    "o": "0.0010", // 24小时前开始第一笔成交价格
}
```

```

    "h": "0.0025",           // 24小时内最高成交价
    "l": "0.0010",           // 24小时内最低成交价
    "v": "10000",            // 成交量
    "q": "18"                // 成交额
}

```

按Symbol刷新的最近24小时精简ticker信息

Stream 名称: <symbol>@miniTicker

Update Speed: 1000ms

全市场所有Symbol的精简Ticker

Payload:

```
[
{
    // 数组每一个元素对应一个交易对，内容与 \<symbol\>@miniTicker相同
}
]
```

同上，只是推送所有交易对.需要注意的是，只有更新的ticker才会被推送.

Stream名称: !miniTicker@arr

Update Speed: 1000ms

按Symbol的完整Ticker

Payload:

```
{
    "e": "24hrTicker",      // 事件类型
    "E": 123456789,        // 事件时间
    "s": "BNBBTC",          // 交易对
    "p": "0.0015",          // 24小时价格变化
    "P": "250.00",          // 24小时价格变化(百分比)
    "w": "0.0018",          // 平均价格
    "x": "0.0009",          // 整整24小时之前，向前数的最后一次成交价格
    "c": "0.0025",          // 最新成交价格
    "Q": "10",               // 最新成交交易的成交量
    "b": "0.0024",          // 目前最高买单价
    "B": "10",               // 目前最高买单价的挂单量
    "a": "0.0026",          // 目前最低卖单价
    "A": "100",              // 目前最低卖单价的挂单量
    "o": "0.0010",          // 整整24小时前，向后数的第一次成交价格
    "h": "0.0025",          // 24小时内最高成交价
    "l": "0.0010",          // 24小时内最低成交价
    "v": "10000",            // 24小时内成交量
    "q": "18",                // 24小时内成交额
}
```

```

    "O": 0,          // 统计开始时间
    "C": 86400000,  // 统计结束时间
    "F": 0,          // 24小时内第一笔成交交易ID
    "L": 18150,     // 24小时内最后一笔成交交易ID
    "n": 18151      // 24小时内成交数
}

```

每秒推送单个交易对的过去24小时滚动窗口标签统计信息。

Stream Name: <symbol>@ticker

Update Speed: 1000ms

全市场所有交易对的完整Ticker

Payload:

```
[
{
    // Same as <symbol>@ticker payload
}
]
```

Stream Name: !ticker@arr

Update Speed: 1000ms

推送全市场所有交易对刷新的24小时完整ticker信息。需要注意的是，没有更新的ticker不会被推送。

按Symbol的最优挂单信息

Payload:

```
{
    "u":400900217,      // order book updateId
    "s":"BNBUSDT",       // 交易对
    "b":"25.35190000",  // 买单最优挂单价格
    "B":"31.21000000",  // 买单最优挂单数量
    "a":"25.36520000",  // 卖单最优挂单价格
    "A":"40.66000000"   // 卖单最优挂单数量
}
```

实时推送指定交易对最优挂单信息

多个 <symbol>@bookTicker 可以订阅在一个WebSocket连接上.

Stream Name: <symbol>@bookTicker

Update Speed: 实时

全市场最优挂单信息

Payload:

```
{
    // 同 <symbol>@bookTicker payload
}
```

实时推送所有交易对最优挂单信息

❶ 这个功能计划在2022年11月前后下线。此推送下线后，可以使用<symbol>@bookTicker来获得单symbol最优挂单信息。可以在一个连接上订阅多个<symbol>@bookTicker。

Stream Name: !bookTicker

Update Speed: 实时

有限档深度信息

Payload:

```
{
    "lastUpdateId": 160,      // Last update ID
    "bids": [                  // Bids to be updated
        [
            "0.0024",           // Price level to be updated
            "10"                // Quantity
        ]
    ],
    "asks": [                  // Asks to be updated
        [
            "0.0026",           // Price level to be updated
            "100"               // Quantity
        ]
    ]
}
```

每秒或每100毫秒推送有限档深度信息。levels表示几档买卖单信息，可选 5/10/20 档

Stream Names: <symbol>@depth<levels> 或 <symbol>@depth<levels>@100ms.

Update Speed: 1000ms 或 100ms

增量深度信息

Payload:

```
{
  "e": "depthUpdate", // 事件类型
  "E": 123456789, // 事件时间
  "s": "BNBBTC", // 交易对
  "U": 157, // 从上次推送至今新增的第一个 update Id
  "u": 160, // 从上次推送至今新增的最后一个 update Id
  "b": [ // 变动的买单深度
    [
      "0.0024", // 变动的价格档位
      "10" // 数量
    ]
  ],
  "a": [ // 变动的卖单深度
    [
      "0.0026", // 变动的价格档位
      "100" // 数量
    ]
  ]
}
```

每秒或每100毫秒推送orderbook的变化部分(如果有)

Stream Name: <symbol>@depth 或 <symbol>@depth@100ms

Update Speed: 1000ms 或 100ms

按Symbol的滚动窗口统计

Payload:

```
{
  "e": "1hTicker", // Event type
  "E": 123456789, // Event time
  "s": "BNBBTC", // Symbol
  "p": "0.0015", // Price change
  "P": "250.00", // Price change percent
  "o": "0.0010", // Open price
  "h": "0.0025", // High price
  "l": "0.0010", // Low price
  "c": "0.0025", // Last price
  "w": "0.0018", // Weighted average price
  "v": "10000", // Total traded base asset volume
  "q": "18", // Total traded quote asset volume
  "O": 0, // Statistics open time
  "C": 86400000, // Statistics close time
  "F": 0, // First trade ID
  "L": 18150, // Last trade Id
  "n": 18151 // Total number of trades
}
```

单个symbol的滚动窗口统计, 支持多个时间窗口。

Stream 名称: <symbol>@ticker_<>window_size>

Window Sizes: 1h, 4h, 1d**更新速度:** 1000ms**注意:**

- 该数据流和 `<symbol>@ticker` 不一样。 - `0 (open time)` 会在每分钟整点开始，而 `0 (closing time)` 是当前更新时间。 - 实际统计的时间范围会比 `<window_size>` 多不超过 59999ms。

全市场滚动窗口统计

Payload:

```
[
  {
    // 同 <symbol>@ticker_<window-size> payload,
    // 间隔内更新的每个symbol。
  }
]
```

全市场symbols的滚动窗口ticker统计，计算于多个窗口。

注意：有变动的ticker才会推送。

Stream 名称: `!ticker_<window-size>@arr`**Window Size:** 1h, 4h, 1d**更新速度:** 1000ms

如何正确在本地维护一个orderbook副本

1. 订阅 `wss://stream.binance.com:9443/ws/bnbbtc@depth`
2. 开始缓存收到的更新。同一个价位，后收到的更新覆盖前面的。
3. 访问Rest接口 `https://api.binance.com/api/v3/depth?symbol=BNBBTC&limit=1000` 获得一个1000档的深度快照
4. 将目前缓存到的信息中 `0 <= u <= 步骤3中获取到的快照中的lastUpdateId` 的部分丢弃(丢弃更早的信息，已经过期)。
5. 将深度快照中的内容更新到本地orderbook副本中，并从websocket接收到的第一个 `0 <= lastUpdateId + 1 且 0 >= lastUpdateId + 1` 的event开始继续更新本地副本。
6. 每一个新event的 `0` 应该恰好等于上一个event的 `0 + 1`，否则可能出现了丢包，请从step3重新进行初始化。
7. 每一个event中的挂单量代表这个价格目前的挂单量**绝对值**，而不是相对变化。
8. 如果某个价格对应的挂单量为0，表示该价位的挂单已经撤单或者被吃，应该移除这个价位。

注意: 因为深度快照对价格档位数量有限制，初始快照之外的价格档位并且没有数量变化的价格档位不会出现在增量深度的更新信息内。因此，即使应用来自增量深度的所有更新，这些价格档位也不会在本地 order book 中可见，所以本地的 order book 与真实的 order book 可能会有一些差异。不过对于大多数用例，5000 的深度限制足以有效地了解市场和交易。

现货账户和交易接口

测试下单 (TRADE)

响应

```
{}
```

`POST /api/v3/order/test (HMAC SHA256)`

用于测试订单请求，但不会提交到撮合引擎

权重: 1

参数:

同于 `POST /api/v3/order`

数据源: 缓存

下单 (TRADE)

Response ACK:

```
{
  "symbol": "BTCUSDT",
  "orderId": 28,
  "orderListId": -1, // OCO订单ID，否则为 -1
  "clientOrderId": "6gCrw2kRUAf9CvJDGP16IP",
  "transactTime": 1507725176595
}
```

Response RESULT:

```
{
  "symbol": "BTCUSDT",
  "orderId": 28,
  "orderListId": -1, // OCO订单ID，否则为 -1
  "clientOrderId": "6gCrw2kRUAf9CvJDGP16IP",
  "transactTime": 1507725176595,
  "price": "0.00000000",
  "origQty": "10.00000000",
  "executedQty": "10.00000000",
  "cummulativeQuoteQty": "10.00000000",
  "status": "FILLED",
  "timeInForce": "GTC",
  "type": "MARKET",
  "side": "SELL",
  "strategyId": 1,           // 下单填了参数才会返回
  "strategyType": 1000000    // 下单填了参数才会返回
}
```

Response FULL:

```
{
  "symbol": "BTCUSDT", // 交易对
  "orderId": 28, // 系统的订单ID
```

```

"orderListId": -1, // OCO订单ID, 否则为 -1
"clientOrderId": "6gCrw2kRUAf9CvJDGP16IP", // 客户自己设置的ID
"transactTime": 1507725176595, // 交易的时间戳
"price": "0.00000000", // 订单价格
"origQty": "10.00000000", // 用户设置的原始订单数量
"executedQty": "10.00000000", // 交易的订单数量
"cummulativeQuoteQty": "10.00000000", // 累计交易的金额
"status": "FILLED", // 订单状态
"timeInForce": "GTC", // 订单的时效方式
"type": "MARKET", // 订单类型, 比如市价单, 现价单等
"side": "SELL", // 订单方向, 买还是卖
"strategyId": 1, // 下单填了参数才会返回
"strategyType": 1000000 // 下单填了参数才会返回
"fills": [ // 订单中交易的信息
{
    "price": "4000.00000000", // 交易的价格
    "qty": "1.00000000", // 交易的数量
    "commission": "4.00000000", // 手续费金额
    "commissionAsset": "USDT", // 手续费的币种
    "tradeId": 56 // 交易ID
},
{
    "price": "3999.00000000",
    "qty": "5.00000000",
    "commission": "19.99500000",
    "commissionAsset": "USDT",
    "tradeId": 57
},
{
    "price": "3998.00000000",
    "qty": "2.00000000",
    "commission": "7.99600000",
    "commissionAsset": "USDT",
    "tradeId": 58
},
{
    "price": "3997.00000000",
    "qty": "1.00000000",
    "commission": "3.99700000",
    "commissionAsset": "USDT",
    "tradeId": 59
},
{
    "price": "3995.00000000",
    "qty": "1.00000000",
    "commission": "3.99500000",
    "commissionAsset": "USDT",
    "tradeId": 60
}
]
}

```

`POST /api/v3/order (HMAC SHA256)`

发送下单。

权重(UUID): 1 权重(IP): 1

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
side	ENUM	YES	详见枚举定义：订单方向
type	ENUM	YES	详见枚举定义：订单类型
timeInForce	ENUM	NO	详见枚举定义：有效方式
quantity	DECIMAL	NO	
quoteOrderQty	DECIMAL	NO	
price	DECIMAL	NO	
newClientOrderId	STRING	NO	客户自定义的唯一订单ID。如果未发送，则自动生成
stopPrice	DECIMAL	NO	仅 <code>STOP_LOSS</code> , <code>STOP_LOSS_LIMIT</code> , <code>TAKE_PROFIT</code> , 和 <code>TAKE_PROFIT_LIMIT</code> 需要此参数。
trailingDelta	LONG	NO	用于 <code>STOP_LOSS</code> , <code>STOP_LOSS_LIMIT</code> , <code>TAKE_PROFIT</code> , 和 <code>TAKE_PROFIT_LIMIT</code> 类型的订单。更多追踪止盈止损订单细节，请参考 追踪止盈止损(Trailing Stop)订单常见问题
icebergQty	DECIMAL	NO	仅使用 <code>LIMIT</code> , <code>STOP_LOSS_LIMIT</code> , 和 <code>TAKE_PROFIT_LIMIT</code> 创建新的 iceberg 订单时需要此参数
newOrderRespType	ENUM	NO	设置响应JSON。ACK, RESULT或FULL；"MARKET"和"LIMIT"订单类型默认为"FULL"，所有其他订单默认为"ACK"。
strategyId	INT	NO	
strategyType	INT	NO	不能低于 <code>1000000</code>
recvWindow	LONG	NO	赋值不能大于 <code>60000</code>
timestamp	LONG	YES	

基于订单 `type` 不同，强制要求某些参数：

类型	强制要求的参数
<code>LIMIT</code>	<code>timeInForce</code> , <code>quantity</code> , <code>price</code>
<code>MARKET</code>	<code>quantity</code> or <code>quoteOrderQty</code>
<code>STOP_LOSS</code>	<code>quantity</code> , <code>stopPrice</code> 或者 <code>trailingDelta</code>
<code>STOP_LOSS_LIMIT</code>	<code>timeInForce</code> , <code>quantity</code> , <code>price</code> , <code>stopPrice</code> 或者 <code>trailingDelta</code>
<code>TAKE_PROFIT</code>	<code>quantity</code> , <code>stopPrice</code> 或者 <code>trailingDelta</code>
<code>TAKE_PROFIT_LIMIT</code>	<code>timeInForce</code> , <code>quantity</code> , <code>price</code> , <code>stopPrice</code> 或者 <code>trailingDelta</code>
<code>LIMIT_MAKER</code>	<code>quantity</code> , <code>price</code>

其他信息:

- `LIMIT_MAKER`是`LIMIT`订单，如果它们立即匹配并成为吃单方将被拒绝。
- 当触发`stopPrice`时，`STOP_LOSS`和`TAKE_PROFIT`将执行`MARKET`订单。
- 任何`LIMIT`或`LIMIT_MAKER`类型的订单都可以通过发送`icebergQty`而成为`iceberg`订单。
- 任何带有`icebergQty`的订单都必须将`timeInForce`设置为`GTC`。
- 使用`quantity`的市价单`MARKET`明确的是用户想用市价单买入或卖出的数量。
 - 比如在`BTCUSDT`上下一个市价单`quantity`用户指明能够买进或者卖出多少BTC。
- 使用`quoteOrderQty`的市价单`MARKET`明确的是通过买入(或卖出)想要花费(或获取)的报价资产数量; 此时的正确报单数量将会以市场流动性和`quoteOrderQty`被计算出来。
 - 以`BTCUSDT`为例，`quoteOrderQty=100`:
 - 下买单的时候，订单会尽可能的买进价值100USDT的BTC。
 - 下卖单的时候，订单会尽可能的卖出价值100USDT的BTC。
- 使用`quoteOrderQty`的市价单`MARKET`不会突破`LOT_SIZE`的限制规则; 报单会按给定的`quoteOrderQty`尽可能接近地被执行。
- 除非之前的订单已经成交，不然设置了相同的`newClientOrderId`订单会被拒绝。
- 对于`STOP_LOSS`, `STOP_LOSS_LIMIT`, `TAKE_PROFIT_LIMIT`和`TAKE_PROFIT`订单，`trailingDelta`可以和`stopPrice`一起使用。

MARKET版本和LIMIT版本针对市场价格触发订单价格规则：

- 价格高于市价：`止损``买入`, `获利``卖出`
- 价格低于市价：`止损``卖出`, `获利``买入`

关于`newOrderRespType`的三种选择

- **Response ACK:** 返回速度最快，不包含成交信息，信息量最少
- **Response RESULT:** 返回速度居中，返回吃单成交的少量信息
- **Response FULL:** 返回速度最慢，返回吃单成交的详细信息

数据源: 撮合引擎

撤销订单 (TRADE)

响应

```
{
  "symbol": "LTCBTC",
  "origClientOrderId": "myOrder1",
  "orderId": 4,
  "orderListId": -1, // OCO订单ID，否则为 -1
  "clientOrderId": "cancelMyOrder1",
  "price": "2.00000000",
  "origQty": "1.00000000",
  "executedQty": "0.00000000",
  "cummulativeQuoteQty": "0.00000000",
  "status": "CANCELED",
  "timeInForce": "GTC",
  "type": "LIMIT",
  "side": "BUY"
}
```

`DELETE /api/v3/order (HMAC SHA256)`

取消有效订单。

权重(IP): 1

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
orderId	LONG	NO	
origClientOrderId	STRING	NO	
newClientOrderId	STRING	NO	用户自定义的本次撤销操作的ID(注意不是被撤销的订单的自定义ID)。如无指定会自动赋值。
recvWindow	LONG	NO	赋值不得大于 60000
timestamp	LONG	YES	

`orderId` 或 `origClientOrderId` 必须至少发送一个

撤销单一交易对的所有挂单 (TRADE)

Response:

```
[
  {
    "symbol": "BTCUSDT",
    "origClientOrderId": "E6APeyTJvkMvLMYMyqu1KQ4",
    "orderId": 11,
    "orderListId": -1,
    "clientOrderId": "pXLV6Hz6mprAcVYpVMTGgx",
    "price": "0.089853",
    "origQty": "0.178622",
    "executedQty": "0.000000",
    "cumulativeQuoteQty": "0.000000",
    "status": "CANCELED",
    "timeInForce": "GTC",
    "type": "LIMIT",
    "side": "BUY"
  },
  {
    "symbol": "BTCUSDT",
    "origClientOrderId": "A3EF2HCwxgZPFMrfwbgrhv",
    "orderId": 13,
    "orderListId": -1,
    "clientOrderId": "pXLV6Hz6mprAcVYpVMTGgx",
    "price": "0.090430",
    "origQty": "0.178622",
    "executedQty": "0.000000",
    "cumulativeQuoteQty": "0.000000",
    "status": "CANCELED",
    "timeInForce": "GTC",
    "type": "LIMIT",
    "side": "BUY"
  }
]
```

```
{
  "orderListId": 1929,
  "contingencyType": "OCO",
  "listStatusType": "ALL_DONE",
  "listOrderStatus": "ALL_DONE",
  "listClientOrderId": "2inzWQdDvZLHbbAmAozX2N",
  "transactionTime": 1585230948299,
  "symbol": "BTCUSDT",
  "orders": [
    {
      "symbol": "BTCUSDT",
      "orderId": 20,
      "clientOrderId": "CwOOIPHSmYywx6jZX77TdL"
    },
    {
      "symbol": "BTCUSDT",
      "orderId": 21,
      "clientOrderId": "461cPg51vQjV3zIMOXNz39"
    }
  ],
  "orderReports": [
    {
      "symbol": "BTCUSDT",
      "origClientOrderId": "CwOOIPHSmYywx6jZX77TdL",
      "orderId": 20,
      "orderListId": 1929,
      "clientOrderId": "pXLV6Hz6mprAcVYpVMTGgx",
      "price": "0.668611",
      "origQty": "0.690354",
      "executedQty": "0.000000",
      "cummulativeQuoteQty": "0.000000",
      "status": "CANCELED",
      "timeInForce": "GTC",
      "type": "STOP_LOSS_LIMIT",
      "side": "BUY",
      "stopPrice": "0.378131",
      "icebergQty": "0.017083"
    },
    {
      "symbol": "BTCUSDT",
      "origClientOrderId": "461cPg51vQjV3zIMOXNz39",
      "orderId": 21,
      "orderListId": 1929,
      "clientOrderId": "pXLV6Hz6mprAcVYpVMTGgx",
      "price": "0.008791",
      "origQty": "0.690354",
      "executedQty": "0.000000",
      "cummulativeQuoteQty": "0.000000",
      "status": "CANCELED",
      "timeInForce": "GTC",
      "type": "LIMIT_MAKER",
      "side": "BUY",
      "icebergQty": "0.639962"
    }
  ]
}
```

DELETE /api/v3/openOrders

撤销单一交易对下所有挂单, 包括OCO的挂单。

权重(IP): 1

参数:

Name	Type	Mandatory	Description
symbol	STRING	YES	
recvWindow	LONG	NO	不能大于 60000
timestamp	LONG	YES	

数据源: 撮合引擎

撤消挂单再下单 (TRADE)

Response SUCCESS:

```
// 撤单和下单都成功
{
  "cancelResult": "SUCCESS",
  "newOrderResult": "SUCCESS",
  "cancelResponse": {
    "symbol": "BTCUSDT",
    "origClientOrderId": "DnLo3vTAQcjha43lAZhZ0y",
    "orderId": 9,
    "orderListId": -1,
    "clientOrderId": "osxN3JXAtJvKvCqGeMwMVR",
    "price": "0.01000000",
    "origQty": "0.000100",
    "executedQty": "0.00000000",
    "cumulativeQuoteQty": "0.00000000",
    "status": "CANCELED",
    "timeInForce": "GTC",
    "type": "LIMIT",
    "side": "SELL"
  },
  "newOrderResponse": {
    "symbol": "BTCUSDT",
    "orderId": 10,
    "orderListId": -1,
    "clientOrderId": "wOceeeOzN0RyLiQfw7jd8S",
    "transactTime": 1652928801803,
    "price": "0.02000000",
    "origQty": "0.040000",
    "executedQty": "0.00000000",
    "cumulativeQuoteQty": "0.00000000",
    "status": "NEW",
    "timeInForce": "GTC",
    "type": "LIMIT",
    "side": "BUY"
  }
}
```

```

    "fills": []
}
}

```

**** 选择了STOP_ON_FAILURE, 撤单出现错误****

```

{
  "code": -2022,
  "msg": "Order cancel-replace failed.",
  "data": {
    "cancelResult": "FAILURE",
    "newOrderResult": "NOT_ATTEMPTED",
    "cancelResponse": {
      "code": -2011,
      "msg": "Unknown order sent."
    },
    "newOrderResponse": null
  }
}

```

响应：撤单成功，下单失败

```

{
  "code": -2021,
  "msg": "Order cancel-replace partially failed.",
  "data": {
    "cancelResult": "SUCCESS",
    "newOrderResult": "FAILURE",
    "cancelResponse": {
      "symbol": "BTCUSDT",
      "origClientOrderId": "86M8erehfExV8z2RC8Zo8k",
      "orderId": 3,
      "orderListId": -1,
      "clientOrderId": "G1kLo6aDv2KGNTFcjfTSFq",
      "price": "0.006123",
      "origQty": "10000.000000",
      "executedQty": "0.000000",
      "cummulativeQuoteQty": "0.000000",
      "status": "CANCELED",
      "timeInForce": "GTC",
      "type": "LIMIT_MAKER",
      "side": "SELL"
    },
    "newOrderResponse": {
      "code": -2010,
      "msg": "Order would immediately match and take."
    }
  }
}

```

选择ALLOW_FAILURE, 撤单出现错误

```

{
  "code": -2021,
  "msg": "Order cancel-replace partially failed."
}

```

```

"data": {
  "cancelResult": "FAILURE",
  "newOrderResult": "SUCCESS",
  "cancelResponse": {
    "code": -2011,
    "msg": "Unknown order sent."
  },
  "newOrderResponse": {
    "symbol": "BTCUSDT",
    "orderId": 11,
    "orderListId": -1,
    "clientOrderId": "pfojJMg6IMNDKuJqDxvoxN",
    "transactTime": 1648540168818
  }
}
}
}

```

响应 : 撤单和下单失败

```

{
  "code": -2022,
  "msg": "Order cancel-replace failed.",
  "data": {
    "cancelResult": "FAILURE",
    "newOrderResult": "FAILURE",
    "cancelResponse": {
      "code": -2011,
      "msg": "Unknown order sent."
    },
    "newOrderResponse": {
      "code": -2010,
      "msg": "Order would immediately match and take."
    }
  }
}

```

`POST /api/v3/order/cancelReplace`

撤消挂单并在同个交易对上重新下单。

在撤消订单和下单前会判断: 1) 过滤器参数, 以及 2) 目前下单数量。

即使请求中没有尝试发送新订单, 比如(`newOrderResult: NOT_ATTEMPTED`), 下单的数量仍然会加1。

Weight(IP): 1

Parameters:

Name	Type	Mandatory	Description
symbol	STRING	YES	
side	ENUM	YES	
type	ENUM	YES	
cancelReplaceMode	ENUM	YES	指定类型 : <code>STOP_ON_FAILURE</code> - 如果撤消订单失败将不会继续重新下单。 <code>ALLOW_FAILURE</code> - 不管撤消订单是否成功都会继续重新下单。

Name	Type	Mandatory	Description
timeInForce	ENUM	NO	
quantity	DECIMAL	NO	
quoteOrderQty	DECIMAL	NO	
price	DECIMAL	NO	
cancelNewClientOrderId	STRING	NO	用户自定义的id，如空缺系统会自动赋值
cancelOrigClientOrderId	STRING	NO	必须提供 <code>cancelOrigClientOrderId</code> 或者 <code>cancelOrderId</code> 。如果两个参数都提供, <code>cancelOrderId</code> 会占优先。
cancelOrderId	LONG	NO	必须提供 <code>cancelOrigClientOrderId</code> 或者 <code>cancelOrderId</code> 。如果两个参数都提供, <code>cancelOrderId</code> 会占优先。
newClientOrderId	STRING	NO	用于辨识新订单。
strategyId	INT	NO	
strategyType	INT	NO	不能低于 <code>1000000</code>
stopPrice	DECIMAL	NO	
trailingDelta	LONG	NO	
icebergQty	DECIMAL	NO	
newOrderRespType	ENUM	NO	指定响应类型: 指定响应类型 <code>ACK</code> , <code>RESULT</code> , or <code>FULL</code> ; <code>MARKET</code> 与 <code>LIMIT</code> 订单默认为 <code>FULL</code> , 其他默认为 <code>ACK</code> .
recvWindow	LONG	NO	不能大于 <code>60000</code>
timestamp	LONG	YES	

如同 `POST /api/v3/order` , 额外的强制参数取决于 `type` 。

响应格式根据消息的处理是成功、部分成功还是失败而有所不同。

数据来源: 撮合引擎

查询订单 (USER_DATA)

响应

```
{
  "symbol": "LTCBTC", // 交易对
  "orderId": 1, // 系统的订单ID
  "orderListId": -1, // OCO订单的ID, 不然就是-1
  "clientOrderId": "myOrder1", // 客户自己设置的ID
  "price": "0.1", // 订单价格
  "origQty": "1.0", // 用户设置的原始订单数量
  "executedQty": "0.0", // 交易的订单数量
}
```

```

    "cummulativeQuoteQty": "0.0", // 累计交易的金额
    "status": "NEW", // 订单状态
    "timeInForce": "GTC", // 订单的时效方式
    "type": "LIMIT", // 订单类型, 比如市价单, 现价单等
    "side": "BUY", // 订单方向, 买还是卖
    "stopPrice": "0.0", // 止损价格
    "icebergQty": "0.0", // 冰山数量
    "time": 1499827319559, // 订单时间
    "updateTime": 1499827319559, // 最后更新时间
    "isWorking": true, // 订单是否出现在orderbook中
    "origQuoteOrderQty": "0.000000" // 原始的交易金额
}

```

GET /api/v3/order (HMAC SHA256)

查询订单状态。

权重(IP): 2

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
orderId	LONG	NO	
origClientOrderId	STRING	NO	
recvWindow	LONG	NO	赋值不得大于 60000
timestamp	LONG	YES	

注意:

- 至少需要发送 `orderId` 与 `origClientOrderId` 中的一个
- 某些订单中 `cummulativeQuoteQty` < 0, 是由于这些订单是 `cummulativeQuoteQty` 功能上线之前的订单。

数据源: 数据库

当前挂单 (USER_DATA)

响应

```
[
{
    "symbol": "LTCBTC",
    "orderId": 1,
    "orderListId": -1, // OCO订单ID, 否则为 -1
    "clientOrderId": "myOrder1",
    "price": "0.1",
    "origQty": "1.0",
    "executedQty": "0.0",
    "cummulativeQuoteQty": "0.0",
    "status": "NEW",
    "timeInForce": "GTC",
}
```

```

    "type": "LIMIT",
    "side": "BUY",
    "stopPrice": "0.0",
    "icebergQty": "0.0",
    "time": 1499827319559,
    "updateTime": 1499827319559,
    "isWorking": true,
    "origQuoteOrderQty": "0.000000"
}
]

```

GET /api/v3/openOrders (HMAC SHA256)

获取交易对的所有当前挂单， 请小心使用不带交易对参数的调用。

权重(IP): 3 单一交易对;

40 交易对参数缺失;

参数:

名称	类型	是否必需	描述
symbol	STRING	NO	
recvWindow	LONG	NO	赋值不得大于 60000
timestamp	LONG	YES	

- 不带symbol参数，会返回所有交易对的挂单

数据源: 数据库

查询所有订单 (USER_DATA)

响应

```

[
{
    "symbol": "LTCBTC",
    "orderId": 1,
    "orderListId": -1, // OCO订单ID, 否则为 -1
    "clientOrderId": "myOrder1",
    "price": "0.1",
    "origQty": "1.0",
    "executedQty": "0.0",
    "cumulativeQuoteQty": "0.0",
    "status": "NEW",
    "timeInForce": "GTC",
    "type": "LIMIT",
    "side": "BUY",
    "stopPrice": "0.0",
    "icebergQty": "0.0",
    "time": 1499827319559,
    "updateTime": 1499827319559,
    "isWorking": true,
}
]

```

```

        "origQuoteOrderQty": "0.000000"
    }
]

```

GET /api/v3/allOrders (HMAC SHA256)

获取所有帐户订单；有效，已取消或已完成。

权重(IP): 10 带有symbol

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
orderId	LONG	NO	
startTime	LONG	NO	
endTime	LONG	NO	
limit	INT	NO	默认 500; 最大 1000.
recvWindow	LONG	NO	赋值不得大于 60000
timestamp	LONG	YES	

注意:

- 如设置 **orderId**，订单量将 \geq **orderId**。否则将返回最新订单。
- 一些历史订单 **cummulativeQuoteQty** < 0, 是指数据此时不存在。
- 如果设置 **startTime** 和 **endTime**, **orderId** 就不需要设置。

数据源: 数据库

OCO下单(TRADE)

响应

```
{
  "orderListId": 0,
  "contingencyType": "OCO",
  "listStatusType": "EXEC_STARTED",
  "listOrderStatus": "EXECUTING",
  "listClientOrderId": "JYVpp3F0f5CAG15Dhtrqlp",
  "transactionTime": 1563417480525,
  "symbol": "LTCBTC",
  "orders": [
    {
      "symbol": "LTCBTC",
      "orderId": 2,
      "clientOrderId": "Kk7sqHb9J6mJwTMDVw7Vos"
    },
    {
      "symbol": "LTCBTC",

```

```

    "orderId": 3,
    "clientOrderId": "xTXKaGYd4bluPVp78IVRv1"
}
],
"orderReports": [
{
    "symbol": "LTCBTC",
    "orderId": 2,
    "orderListId": 0,
    "clientOrderId": "Kk7sqHb9J6mJWTMDVW7Vos",
    "transactTime": 1563417480525,
    "price": "0.000000",
    "origQty": "0.624363",
    "executedQty": "0.000000",
    "cummulativeQuoteQty": "0.000000",
    "status": "NEW",
    "timeInForce": "GTC",
    "type": "STOP_LOSS",
    "side": "BUY",
    "stopPrice": "0.960664"
},
{
    "symbol": "LTCBTC",
    "orderId": 3,
    "orderListId": 0,
    "clientOrderId": "xTXKaGYd4bluPVp78IVRv1",
    "transactTime": 1563417480525,
    "price": "0.036435",
    "origQty": "0.624363",
    "executedQty": "0.000000",
    "cummulativeQuoteQty": "0.000000",
    "status": "NEW",
    "timeInForce": "GTC",
    "type": "LIMIT_MAKER",
    "side": "BUY"
}
]
}
]
}

```

POST /api/v3/order/oco (HMAC SHA256)

发送新 OCO 订单。

权重(UID): 2 权重(IP): 1

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
listClientOrderId	STRING	NO	整个orderList的唯一ID
side	ENUM	YES	详见枚举定义：订单方向
quantity	DECIMAL	YES	
limitClientOrderId	STRING	NO	限价单的唯一ID

名称	类型	是否必需	描述
limitStrategyId	INT	NO	
limitStrategyType	INT	NO	不能低于 1000000
price	DECIMAL	YES	
limitIcebergQty	DECIMAL	NO	
trailingDelta	LONG	NO	
stopClientOrderId	STRING	NO	止损/止损限价单的唯一ID
stopPrice	DECIMAL	YES	
stopStrategyId	INT	NO	
stopStrategyType	INT	NO	不能低于 1000000
stopLimitPrice	DECIMAL	NO	如果提供，须配合提交 stopLimitTimeInForce
stopIcebergQty	DECIMAL	NO	
stopLimitTimeInForce	ENUM	NO	有效值 GTC/FOK/IOC
newOrderRespType	ENUM	NO	详见枚举定义：订单返回类型
recvWindow	LONG	NO	不能大于 60000
timestamp	LONG	YES	

其他信息:

- 价格限制:
 - SELL: 限价 > 最新成交价 > 触发价
 - BUY: 限价 < 最新成交价 < 触发价
- 数量限制:
 - 两个 legs 必须具有同样的数量。
 - ICEBERG 数量不必相同
- 下单rate
 - 一个 OCO 订单被算成2个普通订单.

数据源: 撮合引擎

取消 OCO 订单(TRADE)

Response:

```
{
  "orderId": 0,
  "contingencyType": "OCO",
  "listStatusType": "ALL_DONE",
  "listOrderStatus": "ALL_DONE",
  "listClientOrderId": "C3wyj4WVEktd7u9aVBRXcN",
  "transactionTime": 1574040868128,
  "symbol": "LTCBTC",
}
```

```

"orders": [
  {
    "symbol": "LTCBTC",
    "orderId": 2,
    "clientOrderId": "p09uftiFGg3nw2f0dge0Xa"
  },
  {
    "symbol": "LTCBTC",
    "orderId": 3,
    "clientOrderId": "TX0vglzXuaubXAaENpaRCB"
  }
],
"orderReports": [
  {
    "symbol": "LTCBTC",
    "origClientOrderId": "p09uftiFGg3nw2f0dge0Xa",
    "orderId": 2,
    "orderListId": 0,
    "clientOrderId": "unfwT8ig8i0uj6lPuYLez6",
    "price": "1.00000000",
    "origQty": "10.00000000",
    "executedQty": "0.00000000",
    "cummulativeQuoteQty": "0.00000000",
    "status": "CANCELED",
    "timeInForce": "GTC",
    "type": "STOP_LOSS_LIMIT",
    "side": "SELL",
    "stopPrice": "1.00000000"
  },
  {
    "symbol": "LTCBTC",
    "origClientOrderId": "TX0vglzXuaubXAaENpaRCB",
    "orderId": 3,
    "orderListId": 0,
    "clientOrderId": "unfwT8ig8i0uj6lPuYLez6",
    "price": "3.00000000",
    "origQty": "10.00000000",
    "executedQty": "0.00000000",
    "cummulativeQuoteQty": "0.00000000",
    "status": "CANCELED",
    "timeInForce": "GTC",
    "type": "LIMIT_MAKER",
    "side": "SELL"
  }
]
}

```

DELETE /api/v3/orderList (HMAC SHA256)

取消整个订单列表。

权重(IP): 1

参数

名称	类型	是否必需	描述

名称	类型	是否必需	描述
symbol	STRING	YES	
orderListId	LONG	NO	<code>orderListId</code> 或 <code>listClientOrderId</code> 必须被提供
listClientOrderId	STRING	NO	<code>orderListId</code> 或 <code>listClientOrderId</code> 必须被提供
newClientOrderId	STRING	NO	用户自定义的本次撤销操作的ID(注意不是被撤销的订单的自定义ID)。如无指定会自动赋值。
recvWindow	LONG	NO	不能大于 <code>60000</code>
timestamp	LONG	YES	

其他注意点:

- 取消单个 leg 将取消整个 OCO 订单。

数据源: 撮合引擎

查询 OCO (USER_DATA)

响应

```
{
  "orderListId": 27,
  "contingencyType": "OCO",
  "listStatusType": "EXEC_STARTED",
  "listOrderStatus": "EXECUTING",
  "listClientOrderId": "h2USkA5YQpaXHPIrk96xE",
  "transactionTime": 1565245656253,
  "symbol": "LTCBTC",
  "orders": [
    {
      "symbol": "LTCBTC",
      "orderId": 4,
      "clientOrderId": "qD1gy3kc3Gx0rihm9Y3xwS"
    },
    {
      "symbol": "LTCBTC",
      "orderId": 5,
      "clientOrderId": "ARzZ9I00CPM8i3NhU9Ega"
    }
  ]
}
```

`GET /api/v3/orderList (HMAC SHA256)`

根据提供的可选参数检索特定的OCO。

权重(IP): 2

参数:

名称	类型	是否必需	描述
orderListId	LONG	NO	<code>orderListId</code> 或 <code>origClientOrderId</code> 必须提供一个。
origClientOrderId	STRING	NO	<code>orderListId</code> 或 <code>origClientOrderId</code> 必须提供一个。
recvWindow	LONG	NO	赋值不得大于 <code>60000</code>
timestamp	LONG	YES	

数据源: 数据库

查询所有 OCO (USER_DATA)

响应

```
[
  [
    {
      "orderListId": 29,
      "contingencyType": "OCO",
      "listStatusType": "EXEC_STARTED",
      "listOrderStatus": "EXECUTING",
      "listClientOrderId": "amEEAXryFzFwYF1FeRpUoZ",
      "transactionTime": 1565245913483,
      "symbol": "LTCBTC",
      "orders": [
        {
          "symbol": "LTCBTC",
          "orderId": 4,
          "clientOrderId": "oD7aesZqjEGlZrbtRpy5zB"
        },
        {
          "symbol": "LTCBTC",
          "orderId": 5,
          "clientOrderId": "Jr1h6xir0xgeJOUuYQS7V3"
        }
      ]
    },
    {
      "orderListId": 28,
      "contingencyType": "OCO",
      "listStatusType": "EXEC_STARTED",
      "listOrderStatus": "EXECUTING",
      "listClientOrderId": "hG7hFNxJV6cZy3Ze4AUT4d",
      "transactionTime": 1565245913407,
      "symbol": "LTCBTC",
      "orders": [
        {
          "symbol": "LTCBTC",
          "orderId": 2,
          "clientOrderId": "j6lFOfbmFMRjTYA7rRJ0LP"
        },
        {
          "symbol": "LTCBTC",
          "orderId": 3,
          "clientOrderId": "Jr1h6xir0xgeJOUuYQS7V3"
        }
      ]
    }
  ]
]
```

```

    "orderId": 3,
    "clientOrderId": "z0KCj0dditiLS5ekAFtK81"
}
]
}
]

```

`GET /api/v3/allOrderList (HMAC SHA256)`

根据提供的可选参数检索所有的OCO。

权重(IP): 10

参数

名称	类型	是否必需	描述
fromId	LONG	NO	提供该项后, <code>startTime</code> 和 <code>endTime</code> 都不可提供
startTime	LONG	NO	
endTime	LONG	NO	
limit	INT	NO	默认值: 500; 最大值: 1000
recvWindow	LONG	NO	赋值不能超过 <code>60000</code>
timestamp	LONG	YES	

数据源: 数据库

查询 OCO 挂单 (USER_DATA)

响应

```

[
{
  "orderListId": 31,
  "contingencyType": "OCO",
  "listStatusType": "EXEC_STARTED",
  "listOrderStatus": "EXECUTING",
  "listClientOrderId": "wuB13fmulKj3YjdqWEcsnp",
  "transactionTime": 1565246080644,
  "symbol": "LTCBTC",
  "orders": [
    {
      "symbol": "LTCBTC",
      "orderId": 4,
      "clientOrderId": "r3EH2N76dHfLoSZWIUw1bt"
    },
    {
      "symbol": "LTCBTC",
      "orderId": 5,
      "clientOrderId": "Cv1SnyPD3qhqbjpYEHbd2"
    }
  ]
}

```

```

        ]
    }
]
```

`GET /api/v3/openOrderList (HMAC SHA256)`

权重(IP): 3

参数

名称	类型	是否必需	描述
recvWindow	LONG	NO	赋值不能大于 60000
timestamp	LONG	YES	

数据源: 数据库

账户信息 (USER_DATA)

响应

```
{
  "makerCommission": 15,
  "takerCommission": 15,
  "buyerCommission": 0,
  "sellerCommission": 0,
  "canTrade": true,
  "canWithdraw": true,
  "canDeposit": true,
  "brokered": false,
  "updateTime": 123456789,
  "accountType": "SPOT",
  "balances": [
    {
      "asset": "BTC",
      "free": "4723846.89208129",
      "locked": "0.00000000"
    },
    {
      "asset": "LTC",
      "free": "4763368.68006011",
      "locked": "0.00000000"
    }
  ],
  "permissions": [
    "SPOT"
  ]
}
```

`GET /api/v3/account (HMAC SHA256)`

获取当前账户信息。

权重(IP): 10

参数:

名称	类型	是否必需	描述
recvWindow	LONG	NO	赋值不能大于 60000
timestamp	LONG	YES	

数据源: 缓存 => 数据库

账户成交历史 (USER_DATA)

响应

```
[
  {
    "symbol": "BNBBTC", // 交易对
    "id": 28457, // trade ID
    "orderId": 100234, // 订单ID
    "orderListId": -1, // OCO订单的ID, 不然就是-1
    "price": "4.00000100", // 成交价格
    "qty": "12.00000000", // 成交量
    "quoteQty": "48.000012", // 成交金额
    "commission": "10.10000000", // 交易费金额
    "commissionAsset": "BNB", // 交易费资产类型
    "time": 1499865549590, // 交易时间
    "isBuyer": true, // 是否是买家
    "isMaker": false, // 是否是挂单方
    "isBestMatch": true
  }
]
```

GET /api/v3/myTrades (HMAC SHA256)

获取账户指定交易对的成交历史

权重(IP): 10

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
orderId	LONG	NO	必须要和参数 symbol 一起使用.
startTime	LONG	NO	
endTime	LONG	NO	
fromId	LONG	NO	起始Trade id。 默认获取最新交易。
limit	INT	NO	默认 500; 最大 1000.
recvWindow	LONG	NO	赋值不能超过 60000

名称	类型	是否必需	描述
timestamp	LONG	YES	

注意:

- 如果设定 `fromId`, 获取订单 >= `fromId`. 否则返回最近订单。
- `startTime`和`endTime`设置的时间间隔不能超过24小时。

数据源: 数据库

查询目前下单数 (TRADE)

响应

```
[
  {
    "rateLimitType": "ORDERS",
    "interval": "SECOND",
    "intervalNum": 10,
    "limit": 10000,
    "count": 0
  },
  {
    "rateLimitType": "ORDERS",
    "interval": "DAY",
    "intervalNum": 1,
    "limit": 20000,
    "count": 0
  }
]
```

GET /api/v3/rateLimit/order

获取用户在当前时间区间内的下单总数。

权重(IP): 20**参数:**

名称	类型	是否必需	描述
recvWindow	LONG	NO	赋值不得大于 60000
timestamp	LONG	YES	

数据源: 缓存

杠杆账户和交易接口

全仓杠杆账户划转 (MARGIN)

响应

```
{
    //transaction id
    "tranId": 100000001
}
```

`POST /sapi/v1/margin/transfer (HMAC SHA256)` 执行现货账户与全仓杠杆账户之间的划转

权重(IP): 600

参数:

名称	类型	是否必需	描述
asset	STRING	YES	被划转的资产, 比如, BTC
amount	DECIMAL	YES	划转数量
type	INT	YES	1: 主账户向全仓杠杆账户划转 2: 全仓杠杆账户向主账户划转
recvWindow	LONG	NO	赋值不能大于 <code>60000</code>
timestamp	LONG	YES	

杠杆账户借贷 (MARGIN)

响应

```
{
    //transaction id
    "tranId": 100000001
}
```

`POST /sapi/v1/margin/loan (HMAC SHA256)` 申请借贷。

权重(UID): 3000

参数:

名称	类型	是否必需	描述
asset	STRING	YES	
isIsolated	STRING	NO	是否逐仓杠杆 , "TRUE", "FALSE", 默认 "FALSE"
symbol	STRING	NO	逐仓交易对 , 配合逐仓使用
amount	DECIMAL	YES	
recvWindow	LONG	NO	赋值不能超过 <code>60000</code>

名称	类型	是否必需	描述
timestamp	LONG	YES	

- 如果 `isIsolated = "TRUE"`, 表示逐仓借贷 , 此时 `symbol` 必填
- 如果 `isIsolated = "FALSE"` 表示全仓借贷

杠杆账户归还借贷 (MARGIN)

响应

```
{
  //transaction id
  "tranId": 100000001
}
```

`POST /sapi/v1/margin/repay (HMAC SHA256)` 获取杠杆账户归还借贷。

权重(UID): 3000

参数:

名称	类型	是否必需	描述
asset	STRING	YES	
isIsolated	STRING	NO	是否逐仓杠杆 , "TRUE", "FALSE", 默认 "FALSE"
symbol	STRING	NO	逐仓交易对 , 配合逐仓使用
amount	DECIMAL	YES	
recvWindow	LONG	NO	赋值不能超过 60000
timestamp	LONG	YES	

- 如果 `isIsolated = "TRUE"`, 表示逐仓还款 , 此时 `symbol` 必填
- 如果 `isIsolated = "FALSE"` 表示全仓还款

查询杠杆资产 (MARKET_DATA)

响应

```
{
  "assetFullName": "Binance Coin",
  "assetName": "BNB",
  "isBorrowable": false,
  "isMortgageable": true,
  "userMinBorrow": "0.00000000",
  "userMinRepay": "0.00000000"
}
```

`GET /sapi/v1/margin/asset`

权重(IP): 10

参数:

名称	类型	是否必需	描述
asset	STRING	YES	

查询全仓杠杆交易对 (MARKET_DATA)

响应

```
{
  "id":323355778339572400,
  "symbol":"BTCUSDT",
  "base":"BTC",
  "quote":"USDT",
  "isMarginTrade":true,
  "isBuyAllowed":true,
  "isSellAllowed":true
}
```

`GET /sapi/v1/margin/pair`

权重(IP): 10

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	

获取所有杠杆资产信息 (MARKET_DATA)

响应

```
[
  {
    "assetFullName": "USD coin",
    "assetName": "USDC",
    "isBorrowable": true,
    "isMortgageable": true,
    "userMinBorrow": "0.00000000",
    "userMinRepay": "0.00000000"
  },
  {
    "assetFullName": "BNB-coin",
    "assetName": "BNB",
    "isBorrowable": true,
```

```

    "isMortgageable": true,
    "userMinBorrow": "1.00000000",
    "userMinRepay": "0.00000000"
},
{
    "assetFullName": "Tether",
    "assetName": "USDT",
    "isBorrowable": true,
    "isMortgageable": true,
    "userMinBorrow": "1.00000000",
    "userMinRepay": "0.00000000"
},
{
    "assetFullName": "etherum",
    "assetName": "ETH",
    "isBorrowable": true,
    "isMortgageable": true,
    "userMinBorrow": "0.00000000",
    "userMinRepay": "0.00000000"
},
{
    "assetFullName": "Bitcoin",
    "assetName": "BTC",
    "isBorrowable": true,
    "isMortgageable": true,
    "userMinBorrow": "0.00000000",
    "userMinRepay": "0.00000000"
}
]

```

`GET /sapi/v1/margin/allAssets`

权重(IP): 1

参数:

None

获取所有全仓杠杆交易对(MARKET_DATA)

响应

```

[
{
    "base": "BNB",
    "id": 351637150141315861,
    "isBuyAllowed": true,
    "isMarginTrade": true,
    "isSellAllowed": true,
    "quote": "BTC",
    "symbol": "BNBBTC"
},
{
    "base": "TRX",
    "id": 351637923235429141,

```

```

    "isBuyAllowed": true,
    "isMarginTrade": true,
    "isSellAllowed": true,
    "quote": "BTC",
    "symbol": "TRXBTC"
},
{
    "base": "XRP",
    "id": 351638112213990165,
    "isBuyAllowed": true,
    "isMarginTrade": true,
    "isSellAllowed": true,
    "quote": "BTC",
    "symbol": "XRPBTC"
},
{
    "base": "ETH",
    "id": 351638524530850581,
    "isBuyAllowed": true,
    "isMarginTrade": true,
    "isSellAllowed": true,
    "quote": "BTC",
    "symbol": "ETHBTC"
},
{
    "base": "BNB",
    "id": 376870400832855109,
    "isBuyAllowed": true,
    "isMarginTrade": true,
    "isSellAllowed": true,
    "quote": "USDT",
    "symbol": "BNBUSDT"
}
]

```

`GET /sapi/v1/margin/allPairs`

权重(IP): 1

参数:

None

查询杠杆价格指数 (MARKET_DATA)

响应

```
{
    "calcTime": 1562046418000,
    "price": "0.00333930",
    "symbol": "BNBBTC"
}
```

`GET /sapi/v1/margin/priceIndex`

权重(IP): 10**参数:**

名称	类型	是否必需	描述
symbol	STRING	YES	

杠杆账户下单 (TRADE)

Response ACK:

```
{
  "symbol": "BTCUSDT",
  "orderId": 28,
  "clientOrderId": "6gCrw2kRUAf9CvJDGP16IP",
  "isIsolated": true,           // 是否是逐仓symbol交易
  "transactTime": 1507725176595
}
```

Response RESULT:

```
{
  "symbol": "BTCUSDT",
  "orderId": 28,
  "clientOrderId": "6gCrw2kRUAf9CvJDGP16IP",
  "transactTime": 1507725176595,
  "price": "1.00000000",
  "origQty": "10.00000000",
  "executedQty": "10.00000000",
  "cummulativeQuoteQty": "10.00000000",
  "status": "FILLED",
  "timeInForce": "GTC",
  "type": "MARKET",
  "isIsolated": true,           // 是否是逐仓symbol交易
  "side": "SELL"
}
```

Response FULL:

```
{
  "symbol": "BTCUSDT",
  "orderId": 28,
  "clientOrderId": "6gCrw2kRUAf9CvJDGP16IP",
  "transactTime": 1507725176595,
  "price": "0.00000000",
  "origQty": "10.00000000",
  "executedQty": "10.00000000",
  "cummulativeQuoteQty": "10.00000000",
  "status": "FILLED",
  "timeInForce": "GTC",
  "type": "MARKET",
```

```

"side": "SELL",
"marginBuyBorrowAmount": 5,           // 下单后没有发生借款则不返回该字段
"marginBuyBorrowAsset": "BTC",        // 下单后没有发生借款则不返回该字段
"isIsolated": true,                // 是否是逐仓symbol交易
"fills": [
{
  "price": "4000.00000000",
  "qty": "1.00000000",
  "commission": "4.00000000",
  "commissionAsset": "USDT"
},
{
  "price": "3999.00000000",
  "qty": "5.00000000",
  "commission": "19.99500000",
  "commissionAsset": "USDT"
},
{
  "price": "3998.00000000",
  "qty": "2.00000000",
  "commission": "7.99600000",
  "commissionAsset": "USDT"
},
{
  "price": "3997.00000000",
  "qty": "1.00000000",
  "commission": "3.99700000",
  "commissionAsset": "USDT"
},
{
  "price": "3995.00000000",
  "qty": "1.00000000",
  "commission": "3.99500000",
  "commissionAsset": "USDT"
}
]
}

```

POST /sapi/v1/margin/order (HMAC SHA256)

权重(UID): 6

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
isIsolated	STRING	NO	是否逐仓杠杆 , "TRUE", "FALSE", 默认 "FALSE"
side	ENUM	YES	BUY SELL
type	ENUM	YES	详见枚举定义 : 订单类型
quantity	DECIMAL	NO	
quoteOrderQty	DECIMAL	NO	

名称	类型	是否必需	描述
price	DECIMAL	NO	
stopPrice	DECIMAL	NO	与 <code>STOP_LOSS</code> , <code>STOP_LOSS_LIMIT</code> , <code>TAKE_PROFIT</code> , 和 <code>TAKE_PROFIT_LIMIT</code> 订单一起使用。
newClientOrderId	STRING	NO	客户自定义的唯一订单ID。若未发送自动生成。
icebergQty	DECIMAL	NO	与 <code>LIMIT</code> , <code>STOP_LOSS_LIMIT</code> , 和 <code>TAKE_PROFIT_LIMIT</code> 一起使用创建 iceberg 订单。
newOrderRespType	ENUM	NO	设置响应: JSON, ACK, RESULT, 或 FULL; MARKET 和 LIMIT 订单类型默认为 FULL, 所有其他订单默认为 ACK.
sideEffectType	ENUM	NO	NO_SIDE_EFFECT, MARGIN_BUY, AUTO_REPAY; 默认为 NO_SIDE_EFFECT.
timeInForce	ENUM	NO	GTC,IOC,FOK
recvWindow	LONG	NO	赋值不能大于 <code>60000</code>
timestamp	LONG	YES	

杠杆账户撤销订单 (TRADE)

响应

```
{
  "symbol": "LTCBTC",
  "orderId": 28,
  "origClientOrderId": "myOrder1",
  "clientOrderId": "cancelMyOrder1",
  "price": "1.00000000",
  "origQty": "10.00000000",
  "executedQty": "8.00000000",
  "cummulativeQuoteQty": "8.00000000",
  "status": "CANCELED",
  "timeInForce": "GTC",
  "type": "LIMIT",
  "side": "SELL",
  "isIsolated": true      // 是否是逐仓symbol交易
}
```

`DELETE /sapi/v1/margin/order (HMAC SHA256)`

杠杆账户撤销有效订单。

权重(IP): 10

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	

名称	类型	是否必需	描述
isIsolated	STRING	NO	是否逐仓杠杆 , "TRUE", "FALSE", 默认 "FALSE"
orderId	LONG	NO	
origClientOrderId	STRING	NO	
newClientOrderId	STRING	NO	用于唯一识别此撤销订单 , 默认自动生成。
recvWindow	LONG	NO	T赋值不能大于 60000
timestamp	LONG	YES	

- 必须发送 orderId 或 origClientOrderId 其中一个。

杠杆账户撤销单一交易对的所有挂单 (TRADE)

响应:

```
[
  {
    "symbol": "BTCUSDT",
    "isIsolated": true,          // 是否是逐仓symbol交易
    "origClientOrderId": "E6APeyTJvkMvLMyMqu1KQ4",
    "orderId": 11,
    "orderListId": -1,
    "clientOrderId": "pXLV6Hz6mprAcVYpVMTGgx",
    "price": "0.089853",
    "origQty": "0.178622",
    "executedQty": "0.000000",
    "cumulativeQuoteQty": "0.000000",
    "status": "CANCELED",
    "timeInForce": "GTC",
    "type": "LIMIT",
    "side": "BUY"
  },
  {
    "symbol": "BTCUSDT",
    "isIsolated": false,         // 是否是逐仓symbol交易
    "origClientOrderId": "A3EF2HCwxgZPFMrfwbgrhv",
    "orderId": 13,
    "orderListId": -1,
    "clientOrderId": "pXLV6Hz6mprAcVYpVMTGgx",
    "price": "0.090430",
    "origQty": "0.178622",
    "executedQty": "0.000000",
    "cumulativeQuoteQty": "0.000000",
    "status": "CANCELED",
    "timeInForce": "GTC",
    "type": "LIMIT",
    "side": "BUY"
  },
  {
    "orderListId": 1929,
```

```

    "contingencyType": "OCO",
    "listStatusType": "ALL_DONE",
    "listOrderStatus": "ALL_DONE",
    "listClientOrderId": "2inzWQdDvZLHbbAmAozX2N",
    "transactionTime": 1585230948299,
    "symbol": "BTCUSDT",
    "isIsolated": true,           // 是否是逐仓symbol交易
    "orders": [
        {
            "symbol": "BTCUSDT",
            "orderId": 20,
            "clientOrderId": "CwOOIPHSmYywx6jZX77TdL"
        },
        {
            "symbol": "BTCUSDT",
            "orderId": 21,
            "clientOrderId": "461cPg51vQjV3zIMOXNz39"
        }
    ],
    "orderReports": [
        {
            "symbol": "BTCUSDT",
            "origClientOrderId": "CwOOIPHSmYywx6jZX77TdL",
            "orderId": 20,
            "orderListId": 1929,
            "clientOrderId": "pXLV6Hz6mprAcVYpVMTGgx",
            "price": "0.668611",
            "origQty": "0.690354",
            "executedQty": "0.000000",
            "cummulativeQuoteQty": "0.000000",
            "status": "CANCELED",
            "timeInForce": "GTC",
            "type": "STOP_LOSS_LIMIT",
            "side": "BUY",
            "stopPrice": "0.378131",
            "icebergQty": "0.017083"
        },
        {
            "symbol": "BTCUSDT",
            "origClientOrderId": "461cPg51vQjV3zIMOXNz39",
            "orderId": 21,
            "orderListId": 1929,
            "clientOrderId": "pXLV6Hz6mprAcVYpVMTGgx",
            "price": "0.008791",
            "origQty": "0.690354",
            "executedQty": "0.000000",
            "cummulativeQuoteQty": "0.000000",
            "status": "CANCELED",
            "timeInForce": "GTC",
            "type": "LIMIT_MAKER",
            "side": "BUY",
            "icebergQty": "0.639962"
        }
    ]
}
]
}

```

DELETE /sapi/v1/margin/openOrders (HMAC SHA256)

杠杆账户撤销单一交易对下所有挂单, 包括OCO的挂单。

权重(IP): 1

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
isIsolated	STRING	NO	是否逐仓杠杆 , "TRUE", "FALSE", 默认 "FALSE"
recvWindow	LONG	NO	赋值不能大于 60000
timestamp	LONG	YES	

获取全仓杠杆划转历史 (USER_DATA)

响应

```
{
  "rows": [
    {
      "amount": "0.10000000",
      "asset": "BNB",
      "status": "CONFIRMED",
      "timestamp": 1566898617,
      "txId": 5240372201,
      "type": "ROLL_IN"
    },
    {
      "amount": "5.00000000",
      "asset": "USDT",
      "status": "CONFIRMED",
      "timestamp": 1566888436,
      "txId": 5239810406,
      "type": "ROLL_OUT"
    },
    {
      "amount": "1.00000000",
      "asset": "EOS",
      "status": "CONFIRMED",
      "timestamp": 1566888403,
      "txId": 5239808703,
      "type": "ROLL_IN"
    }
  ],
  "total": 3
}
```

GET /sapi/v1/margin/transfer (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
asset	STRING	NO	
type	STRING	NO	划转类型: ROLL_IN, ROLL_OUT
startTime	LONG	NO	
endTime	LONG	NO	
current	LONG	NO	当前查询页。从 1开始。默认:1
size	LONG	NO	默认:10 最大:100
archived	STRING	NO	默认: <code>false</code> . 查询6个月以前的数据，需要设为 <code>true</code>
recvWindow	LONG	NO	赋值不能大于 <code>60000</code>
timestamp	LONG	YES	

- 响应返回为降序排列。
- 查询时间范围最大不得超过30天。
- 若`startTime`和`endTime`没传，则默认返回最近7天数据
- 如果想查询6个月以前数据，设置`archived`为`true`。

查询借贷记录 (USER_DATA)

响应

```
{
  "rows": [
    {
      "isolatedSymbol": "BNBUSDT", // 逐仓借贷 返回逐仓symbol; 若是全仓不会返回此字段
      "txId": 12807067523,
      "asset": "BNB",
      "principal": "0.84624403",
      "timestamp": 1555056425000,
      "status": "CONFIRMED" //状态: PENDING (等待执行), CONFIRMED (成功借贷), FAILED (执行失败);
    }
  ],
  "total": 1
}
```

GET /sapi/v1/margin/loan (HMAC SHA256)

权重(IP): 10

参数:

名称	类型	是否必需	描述
asset	STRING	YES	

名称	类型	是否必需	描述
isolatedSymbol	STRING	NO	逐仓symbol
txId	LONG	NO	<code>tranId</code> in POST /sapi/v1/margin/loan
startTime	LONG	NO	
endTime	LONG	NO	
current	LONG	NO	当前查询页。开始值1。默认:1
size	LONG	NO	默认:10 最大:100
archived	STRING	NO	默认: <code>false</code> . 查询6个月以前的数据，需要设为 <code>true</code>
recvWindow	LONG	NO	赋值不能大于 <code>60000</code>
timestamp	LONG	YES	

- 必须发送 `txId` 或 `startTime`, `txId` 优先。
- 响应返回为降序排列。
- 如果发送 `isolatedSymbol`，返回指定逐仓symbol指定asset的借贷记录。
- 查询时间范围最大不得超过30天。
- 若 `startTime` 和 `endTime` 没传，则默认返回最近7天数据
- 如果想查询6个月以前数据，设置 `archived` 为 `true`。

查询还贷记录 (USER_DATA)

响应

```
{
  "rows": [
    {
      "isolatedSymbol": "BNBUSDT", // 逐仓还款 返回逐仓symbol; 若是全仓不会返回此字段
      "amount": "14.00000000", // 还款总额
      "asset": "BNB",
      "interest": "0.01866667", // 支付的利息
      "principal": "13.98133333", // 支付的本金
      "status": "CONFIRMED", // 状态: PENDING (等待执行), CONFIRMED (成功还款), FAILED (执行失败);
      "timestamp": 1563438204000,
      "txId": 2970933056
    }
  ],
  "total": 1
}
```

`GET /sapi/v1/margin/repay (HMAC SHA256)`

权重(IP) 10

参数:

名称	类型	是否必需	描述
----	----	------	----

名称	类型	是否必需	描述
asset	STRING	YES	
isolatedSymbol	STRING	NO	逐仓symbol
txId	LONG	NO	返回 /sapi/v1/margin/repay
startTime	LONG	NO	
endTime	LONG	NO	
current	LONG	NO	当前查询页。开始值 1. 默认:1
size	LONG	NO	默认:10 最大:100
archived	STRING	NO	默认: <code>false</code> . 查询6个月以前的数据，需要设为 <code>true</code>
recvWindow	LONG	NO	赋值不能大于 <code>60000</code>
timestamp	LONG	YES	

- 必须发送 `txId` 或 `startTime` , `txId` 优先。
- 响应返回为降序排列。
- 如果发送 `isolatedSymbol` , 返回指定逐仓symbol指定asset的还贷记录。
- 查询时间范围最大不得超过30天。
- 若 `startTime` 和 `endTime` 没传，则默认返回最近7天数据
- 如果想查询6个月以前数据，设置 `archived` 为 `true`。

获取利息历史 (USER_DATA)

响应

```
{
  "rows": [
    {
      "isolatedSymbol": "BNBUSDT", // 返回逐仓symbol; 若是全仓不会返回此字段
      "asset": "BNB",
      "rawAsset": "BTC", // 逐仓不会返回此字段
      "interest": "0.02414667",
      "interestAccuredTime": 1566813600000,
      "interestRate": "0.01600000",
      "principal": "36.22000000",
      "type": "ON_BORROW"
    }
  ],
  "total": 1
}
```

GET /sapi/v1/margin/interestHistory (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
asset	STRING	NO	
isolatedSymbol	STRING	NO	逐仓symbol
startTime	LONG	NO	
endTime	LONG	NO	
current	LONG	NO	当前查询页。 开始值 1. 默认:1
size	LONG	NO	默认:10 最大:100
archived	STRING	NO	默认: <code>false</code> . 查询6个月以前的数据，需要设为 <code>true</code>
recvWindow	LONG	NO	赋值不能大于 <code>60000</code>
timestamp	LONG	YES	

- 响应返回为降序排列。
- 如果发送isolatedSymbol，返回指定逐仓symbol的记录。
- 查询时间范围最大不得超过30天。
- 若 startTime 和 endTime 没传，则默认返回最近7天数据
- 如果想查询6个月以前数据，设置 archived 为 true。
- 返回的 type 数据有4种类型：
 - PERIODIC 每小时收的利息
 - ON_BORROW 借款的时候第一次收的利息
 - PERIODIC_CONVERTED 每小时收的利息，用BNB抵扣
 - ON_BORROW_CONVERTED 借款的时候第一次收的利息，用BNB抵扣

获取账户强制平仓记录(USER_DATA)

响应

```
{
  "rows": [
    {
      "avgPrice": "0.00388359",
      "executedQty": "31.39000000",
      "orderId": 180015097,
      "price": "0.00388110",
      "qty": "31.39000000",
      "side": "SELL",
      "symbol": "BNBBTC",
      "timeInForce": "GTC",
      "isIsolated": true,           // 是否是逐仓
      "updatedTime": 1558941374745
    }
  ],
  "total": 1
}
```

GET /sapi/v1/margin/forceLiquidationRec (HMAC SHA256)

权重(IP): 1**参数:**

名称	类型	是否必需	描述
startTime	LONG	NO	
endTime	LONG	NO	
isolatedSymbol	STRING	NO	
current	LONG	NO	当前查询页。 开始值 1. 默认:1
size	LONG	NO	默认:10 最大:100
recvWindow	LONG	NO	赋值不能大于 60000
timestamp	LONG	YES	

- 响应返回为降序排列。

查询全仓杠杆账户详情 (USER_DATA)

响应

```
{
  "borrowEnabled": true,
  "marginLevel": "11.64405625",
  "totalAssetOfBtc": "6.82728457",
  "totalLiabilityOfBtc": "0.58633215",
  "totalNetAssetOfBtc": "6.24095242",
  "tradeEnabled": true,
  "transferEnabled": true,
  "userAssets": [
    {
      "asset": "BTC",
      "borrowed": "0.00000000",
      "free": "0.00499500",
      "interest": "0.00000000",
      "locked": "0.00000000",
      "netAsset": "0.00499500"
    },
    {
      "asset": "BNB",
      "borrowed": "201.66666672",
      "free": "2346.50000000",
      "interest": "0.00000000",
      "locked": "0.00000000",
      "netAsset": "2144.83333328"
    },
    {
      "asset": "ETH",
      "borrowed": "0.00000000",
      "free": "0.00000000",
      "interest": "0.00000000",
      "locked": "0.00000000",
      "netAsset": "0.00000000"
    }
  ]
}
```

```

        "interest": "0.00000000",
        "locked": "0.00000000",
        "netAsset": "0.00000000"
    },
    {
        "asset": "USDT",
        "borrowed": "0.00000000",
        "free": "0.00000000",
        "interest": "0.00000000",
        "locked": "0.00000000",
        "netAsset": "0.00000000"
    }
]
}

```

GET /sapi/v1/margin/account (HMAC SHA256)

权重(IP): 10

参数:

名称	类型	是否必需	描述
recvWindow	LONG	NO	赋值不能大于 60000
timestamp	LONG	YES	

查询杠杆账户订单 (USER_DATA)

响应

```

{
    "clientOrderId": "ZwfQzuDIGpceVhKW5DvCmO",
    "cummulativeQuoteQty": "0.00000000",
    "executedQty": "0.00000000",
    "icebergQty": "0.00000000",
    "isWorking": true,
    "orderId": 213205622,
    "origQty": "0.30000000",
    "price": "0.00493630",
    "side": "SELL",
    "status": "NEW",
    "stopPrice": "0.00000000",
    "symbol": "BNBBTC",
    "isIsolated": true,           // 是否是逐仓symbol交易
    "time": 1562133008725,
    "timeInForce": "GTC",
    "type": "LIMIT",
    "updateTime": 1562133008725
}

```

GET /sapi/v1/margin/order (HMAC SHA256)

权重(IP): 10

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
isIsolated	STRING	NO	是否逐仓杠杆 , "TRUE", "FALSE", 默认 "FALSE"
orderId	LONG	NO	
origClientOrderId	STRING	NO	
recvWindow	LONG	NO	赋值不能大于 60000
timestamp	LONG	YES	

- 必须发送 orderId 或 origClientOrderId 其中一个。
- 一些历史订单的 cummulativeQuoteQty < 0, 是指当前数据不存在。

查询杠杆账户挂单记录 (USER_DATA)

响应

```
[
  {
    "clientOrderId": "qhcZw71gAkCCTv0t0k8LUK",
    "cummulativeQuoteQty": "0.00000000",
    "executedQty": "0.00000000",
    "icebergQty": "0.00000000",
    "isWorking": true,
    "orderId": 211842552,
    "origQty": "0.30000000",
    "price": "0.00475010",
    "side": "SELL",
    "status": "NEW",
    "stopPrice": "0.00000000",
    "symbol": "BNBBTC",
    "isIsolated": true,           // 是否是逐仓symbol交易
    "time": 1562040170089,
    "timeInForce": "GTC",
    "type": "LIMIT",
    "updateTime": 1562040170089
  }
]
```

GET /sapi/v1/margin/openOrders (HMAC SHA256)

权重(IP): 10

参数:

名称	类型	是否必需	描述
symbol	STRING	NO	

名称	类型	是否必需	描述
isIsolated	STRING	NO	是否逐仓杠杆 , "TRUE", "FALSE", 默认 "FALSE"
recvWindow	LONG	NO	赋值不能大于 60000
timestamp	LONG	YES	

- 如未发送symbol，返回所有 symbols 订单记录。
- 当返回所有symbols时，针对限速器计数的请求数量等于当前在交易所交易的symbols数量。
- 如果 isIsolated = "TRUE", symbol 为必填

查询杠杆账户的所有订单 (USER_DATA)

响应

```
[
  {
    "clientOrderId": "D2KDy4DIeS56PvkM13f8cP",
    "cummulativeQuoteQty": "0.0000000",
    "executedQty": "0.0000000",
    "icebergQty": "0.0000000",
    "isWorking": false,
    "orderId": 41295,
    "origQty": "5.3100000",
    "price": "0.2250000",
    "side": "SELL",
    "status": "CANCELED",
    "stopPrice": "0.1800000",
    "symbol": "BNBBTC",
    "isIsolated": false,          // 是否是逐仓symbol交易
    "time": 1565769338806,
    "timeInForce": "GTC",
    "type": "TAKE_PROFIT_LIMIT",
    "updateTime": 1565769342148
  },
  {
    "clientOrderId": "gXYtqhcEAs2Rn9SUD9nRKx",
    "cummulativeQuoteQty": "0.0000000",
    "executedQty": "0.0000000",
    "icebergQty": "1.0000000",
    "isWorking": true,
    "orderId": 41296,
    "origQty": "6.6500000",
    "price": "0.1800000",
    "side": "SELL",
    "status": "CANCELED",
    "stopPrice": "0.0000000",
    "symbol": "BNBBTC",
    "isIsolated": false,          // 是否是逐仓symbol交易
    "time": 1565769348687,
    "timeInForce": "GTC",
    "type": "LIMIT",
    "updateTime": 1565769352226
  }
]
```

}

]

GET /sapi/v1/margin/allOrders (HMAC SHA256)

权重(IP): 200**访问限制**

60次/分钟/IP

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
isIsolated	STRING	NO	是否逐仓杠杆 , "TRUE", "FALSE", 默认 "FALSE"
orderId	LONG	NO	
startTime	LONG	NO	
endTime	LONG	NO	
limit	INT	NO	默认 500;最大500.
recvWindow	LONG	NO	赋值不能大于 60000
timestamp	LONG	YES	

- 如果设置 orderId , 获取订单 >= orderId , 否则返回近期订单历史。
- 一些历史订单的 cummulativeQuoteQty < 0, 是指当前数据不存在。

杠杆账户 OCO 下单(TRADE)**响应**

```
{
  "orderListId": 0,
  "contingencyType": "OCO",
  "listStatusType": "EXEC_STARTED",
  "listOrderStatus": "EXECUTING",
  "listClientOrderId": "JYVpp3F0f5CAG15DhtrqLp",
  "transactionTime": 1563417480525,
  "symbol": "LTCBTC",
  "marginBuyBorrowAmount": "5",           // 下单后没有发生借款则不返回该字段
  "marginBuyBorrowAsset": "BTC",         // 下单后没有发生借款则不返回该字段
  "isIsolated": false,                 // 是否是逐仓symbol交易
  "orders": [
    {
      "symbol": "LTCBTC",
      "orderId": 2,
      "clientOrderId": "Kk7sqHb9J6mJWTMDVW7Vos"
    },
    {
      "symbol": "LTCBTC",
      "orderId": 3,
      "clientOrderId": "Kk7sqHb9J6mJWTMDVW7Vos"
    }
  ]
}
```

```

    "symbol": "LTCBTC",
    "orderId": 3,
    "clientOrderId": "xTXKaGYd4bluPVp78IVRv1"
}
],
"orderReports": [
{
    "symbol": "LTCBTC",
    "orderId": 2,
    "orderListId": 0,
    "clientOrderId": "Kk7sqHb9J6mJWTMDVW7Vos",
    "transactTime": 1563417480525,
    "price": "0.000000",
    "origQty": "0.624363",
    "executedQty": "0.000000",
    "cummulativeQuoteQty": "0.000000",
    "status": "NEW",
    "timeInForce": "GTC",
    "type": "STOP_LOSS",
    "side": "BUY",
    "stopPrice": "0.960664"
},
{
    "symbol": "LTCBTC",
    "orderId": 3,
    "orderListId": 0,
    "clientOrderId": "xTXKaGYd4bluPVp78IVRv1",
    "transactTime": 1563417480525,
    "price": "0.036435",
    "origQty": "0.624363",
    "executedQty": "0.000000",
    "cummulativeQuoteQty": "0.000000",
    "status": "NEW",
    "timeInForce": "GTC",
    "type": "LIMIT_MAKER",
    "side": "BUY"
}
]
}

```

POST /sapi/v1/margin/order/oco (HMAC SHA256)

杠杆账户发送新 OCO 订单。

权重(UID): 6

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
isIsolated	STRING	NO	是否逐仓杠杆 , "TRUE", "FALSE", 默认 "FALSE"
listClientOrderId	STRING	NO	整个orderList的唯一ID
side	ENUM	YES	详见枚举定义 : 订单方向

名称	类型	是否必需	描述
quantity	DECIMAL	YES	
limitClientOrderId	STRING	NO	限价单的唯一ID
price	DECIMAL	YES	
limitIcebergQty	DECIMAL	NO	
stopClientOrderId	STRING	NO	止损/止损限价单的唯一ID
stopPrice	DECIMAL	YES	
stopLimitPrice	DECIMAL	NO	如果提供，须配合提交 stopLimitTimeInForce
stopIcebergQty	DECIMAL	NO	
stopLimitTimeInForce	ENUM	NO	有效值 GTC/FOK/IOC
newOrderRespType	ENUM	NO	详见枚举定义：订单返回类型
sideEffectType	ENUM	NO	NO_SIDE_EFFECT, MARGIN_BUY, AUTO_REPAY; 默认为 NO_SIDE_EFFECT
recvWindow	LONG	NO	不能大于 60000
timestamp	LONG	YES	

其他信息:

- 价格限制:
 - SELL: 限价 > 最新成交价 > 触发价
 - BUY: 限价 < 最新成交价 < 触发价
- 数量限制:
 - 两个 legs 必须具有同样的数量。
 - ICEBERG 数量不必相同
- 下单rate
 - 一个 OCO 订单被算成2个普通订单.

取消杠杆账户 OCO 订单(TRADE)

Response:

```
{
  "orderListId": 0,
  "contingencyType": "OCO",
  "listStatusType": "ALL_DONE",
  "listOrderStatus": "ALL_DONE",
  "listClientOrderId": "C3wyj4WVEktd7u9aVBRXcN",
  "transactionTime": 1574040868128,
  "symbol": "LTCBTC",
  "isIsolated": false, // 是否是逐仓symbol交易
  "orders": [
    {
      "symbol": "LTCBTC",
      "orderId": 2,
      "clientOrderId": "C3wyj4WVEktd7u9aVBRXcN"
    }
  ]
}
```

```

    "clientOrderId": "p09uftiFGg3nw2f0dge0Xa"
  },
  {
    "symbol": "LTCBTC",
    "orderId": 3,
    "clientOrderId": "TX0vglzXuaubXAaENpaRCB"
  }
],
"orderReports": [
  {
    "symbol": "LTCBTC",
    "origClientOrderId": "p09uftiFGg3nw2f0dge0Xa",
    "orderId": 2,
    "orderListId": 0,
    "clientOrderId": "unfWT8ig8i0uj6lPuYLez6",
    "price": "1.00000000",
    "origQty": "10.00000000",
    "executedQty": "0.00000000",
    "cummulativeQuoteQty": "0.00000000",
    "status": "CANCELED",
    "timeInForce": "GTC",
    "type": "STOP_LOSS_LIMIT",
    "side": "SELL",
    "stopPrice": "1.00000000"
  },
  {
    "symbol": "LTCBTC",
    "origClientOrderId": "TX0vglzXuaubXAaENpaRCB",
    "orderId": 3,
    "orderListId": 0,
    "clientOrderId": "unfWT8ig8i0uj6lPuYLez6",
    "price": "3.00000000",
    "origQty": "10.00000000",
    "executedQty": "0.00000000",
    "cummulativeQuoteQty": "0.00000000",
    "status": "CANCELED",
    "timeInForce": "GTC",
    "type": "LIMIT_MAKER",
    "side": "SELL"
  }
]
}

```

DELETE /sapi/v1/margin/orderList (HMAC SHA256)

取消杠杆账户订单列表。

权重(UID): 1

参数

名称	类型	是否必需	描述
symbol	STRING	YES	
isIsolated	STRING	NO	是否逐仓杠杆 , "TRUE", "FALSE", 默认 "FALSE"

名称	类型	是否必需	描述
orderListId	LONG	NO	<code>orderListId</code> 或 <code>listClientOrderId</code> 必须被提供
listClientOrderId	STRING	NO	<code>orderListId</code> 或 <code>listClientOrderId</code> 必须被提供
newClientOrderId	STRING	NO	用户自定义的本次撤销操作的ID(注意不是被撤销的订单的自定义ID)。如无指定会自动赋值。
recvWindow	LONG	NO	不能大于 <code>60000</code>
timestamp	LONG	YES	

其他注意点:

- 取消单个 leg 将取消整个 OCO 订单。

查询杠杆账户 OCO (USER_DATA)

响应

```
{
  "orderListId": 27,
  "contingencyType": "OCO",
  "listStatusType": "EXEC_STARTED",
  "listOrderStatus": "EXECUTING",
  "listClientOrderId": "h2USkA5YQpaXHPIrk96xE",
  "transactionTime": 1565245656253,
  "symbol": "LTCBTC",
  "isIsolated": true,           // 是否是逐仓symbol交易
  "orders": [
    {
      "symbol": "LTCBTC",
      "orderId": 4,
      "clientOrderId": "qD1gy3kc3Gx0rihm9Y3xwS"
    },
    {
      "symbol": "LTCBTC",
      "orderId": 5,
      "clientOrderId": "ARzz9I00CPM8i3NhmU9Ega"
    }
  ]
}
```

GET /sapi/v1/margin/orderList (HMAC SHA256)

根据提供的可选参数检索特定的杠杆账户 OCO 订单。

权重(IP): 10

参数:

名称	类型	是否必需	描述

名称	类型	是否必需	描述
isIsolated	STRING	NO	是否逐仓杠杆 , "TRUE", "FALSE", 默认 "FALSE"
symbol	STRING	NO	逐仓杠杆必填 , 全仓杠杆不支持该参数
orderListId	LONG	NO	<code>orderListId</code> 或 <code>origClientOrderId</code> 必须提供一个。
origClientOrderId	STRING	NO	<code>orderListId</code> 或 <code>origClientOrderId</code> 必须提供一个。
recvWindow	LONG	NO	赋值不得大于 <code>60000</code>
timestamp	LONG	YES	

查询特定杠杆账户所有 OCO (USER_DATA)

响应

```
[
  {
    "orderListId": 29,
    "contingencyType": "OCO",
    "listStatusType": "EXEC_STARTED",
    "listOrderStatus": "EXECUTING",
    "listClientOrderId": "amEEAXryFzFwYF1FeRpUoZ",
    "transactionTime": 1565245913483,
    "symbol": "LTCBTC",
    "isIsolated": true,           // 是否是逐仓symbol交易
    "orders": [
      {
        "symbol": "LTCBTC",
        "orderId": 4,
        "clientOrderId": "oD7aesZqjEG1ZrbtRpy5zB"
      },
      {
        "symbol": "LTCBTC",
        "orderId": 5,
        "clientOrderId": "Jr1h6xir0xgeJOUuYQ57V3"
      }
    ]
  },
  {
    "orderListId": 28,
    "contingencyType": "OCO",
    "listStatusType": "EXEC_STARTED",
    "listOrderStatus": "EXECUTING",
    "listClientOrderId": "hG7hFNxJV6cZy3Ze4AUT4d",
    "transactionTime": 1565245913407,
    "symbol": "LTCBTC",
    "orders": [
      {
        "symbol": "LTCBTC",
        "orderId": 2,
        "clientOrderId": "j6lF0fbmFMRjTYA7rRJ0LP"
      }
    ]
  }
]
```

```

    },
    {
      "symbol": "LTCBTC",
      "orderId": 3,
      "clientOrderId": "z0KCj0dditiLS5ekAFtK81"
    }
  ]
}
]

```

`GET /sapi/v1/margin/allOrderList (HMAC SHA256)`

根据提供的可选参数检索特定杠杆账户所有的 OCO 订单。

权重(IP): 200

参数

名称	类型	是否必需	描述
isIsolated	STRING	NO	是否逐仓杠杆 , "TRUE", "FALSE", 默认 "FALSE"
symbol	STRING	NO	逐仓杠杆必填 , 全仓杠杆不支持该参数
fromId	LONG	NO	提供该项后, <code>startTime</code> 和 <code>endTime</code> 都不可提供
startTime	LONG	NO	
endTime	LONG	NO	
limit	INT	NO	默认值: 500; 最大值: 1000
recvWindow	LONG	NO	赋值不能超过 60000
timestamp	LONG	YES	

查询杠杆账户 OCO 挂单 (USER_DATA)

响应

```

[
  {
    "orderListId": 31,
    "contingencyType": "OCO",
    "listStatusType": "EXEC_STARTED",
    "listOrderStatus": "EXECUTING",
    "listClientOrderId": "wuB13fmulKj3YjdqWEcsnp",
    "transactionTime": 1565246080644,
    "symbol": "LTCBTC",
    "isIsolated": true,           // 是否是逐仓symbol交易
    "orders": [
      {
        "symbol": "LTCBTC",
        "orderId": 4,
        "clientOrderId": "r3EH2N76dHfLoSZWIUw1bT"
      }
    ]
}
]

```

```

    },
    {
      "symbol": "LTCBTC",
      "orderId": 5,
      "clientOrderId": "Cv1SnyPD3qhqpbjpYEHbd2"
    }
  ]
}
]

```

`GET /sapi/v1/margin/openOrderList (HMAC SHA256)`

权重(IP): 10

参数

名称	类型	是否必需	描述
isIsolated	STRING	NO	是否逐仓杠杆 , "TRUE", "FALSE", 默认 "FALSE"
symbol	STRING	NO	逐仓杠杆必填 , 全仓杠杆不支持该参数
recvWindow	LONG	NO	赋值不能大于 60000
timestamp	LONG	YES	

查询杠杆账户交易历史 (USER_DATA)

响应

```

[
  {
    "commission": "0.00006000",
    "commissionAsset": "BTC",
    "id": 34,
    "isBestMatch": true,
    "isBuyer": false,
    "isMaker": false,
    "orderId": 39324,
    "price": "0.02000000",
    "qty": "3.00000000",
    "symbol": "BNBBTC",
    "isIsolated": false, // 是否是逐仓symbol交易
    "time": 1561973357171
  },
  {
    "commission": "0.00002950",
    "commissionAsset": "BTC",
    "id": 32,
    "isBestMatch": true,
    "isBuyer": false,
    "isMaker": true,
    "orderId": 39319,
    "price": "0.00590000",
    "time": 1561973357171
  }
]

```

```

    "qty": "5.00000000",
    "symbol": "BNBBTC",
    "isIsolated": false,      // 是否是逐仓symbol交易
    "time": 1561964645345
}
]

```

`GET /sapi/v1/margin/myTrades (HMAC SHA256)`

权重(IP): 10

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
isIsolated	STRING	NO	是否逐仓杠杆 , "TRUE", "FALSE", 默认 "FALSE"
orderId	LONG	NO	
startTime	LONG	NO	
endTime	LONG	NO	
fromId	LONG	NO	获取TradeId , 默认获取近期交易历史。
limit	INT	NO	默认 500; 最大 1000.
recvWindow	LONG	NO	默认值不能大于 60000
timestamp	LONG	YES	

- 如果设置 fromId , 获取订单 id >= fromId , 否则返回近期订单历史。

查询账户最大可借货额度(USER_DATA)

响应

```
{
  "amount": "1.69248805", // 系统可借充足情况下用户账户当前最大可借额度
  "borrowLimit": "60" // 平台限制的用户当前等级可以借的额度
}
```

`GET /sapi/v1/margin/maxBorrowable (HMAC SHA256)`

权重(IP): 50

参数:

名称	类型	是否必需	描述
asset	STRING	YES	
isolatedSymbol	STRING	NO	逐仓交易对 , 适用于逐仓查询

名称	类型	是否必需	描述
recvWindow	LONG	NO	默认值不能大于 60000
timestamp	LONG	YES	

- `borrowLimit` 的值也可以查看 <https://www.binance.com/cn/margin-fee>

查询最大可转出额 (USER_DATA)

响应

```
{
  "amount": "3.59498107"
}
```

`GET /sapi/v1/margin/maxTransferable (HMAC SHA256)`

权重(IP): 50

参数:

名称	类型	是否必需	描述
asset	STRING	YES	
isolatedSymbol	STRING	NO	逐仓交易对，适用于逐仓查询
recvWindow	LONG	NO	默认值不能大于 60000
timestamp	LONG	YES	

查询Margin账户信息汇总 (USER_DATA)

响应 :

```
{
  "normalBar": "1.5",
  "marginCallBar": "1.3",
  "forceLiquidationBar": "1.1"
}
```

`GET /sapi/v1/margin/tradeCoeff (HMAC SHA256)`

获取用户个人杠杆账户信息汇总

权重 (IP) : 10

参数 :

名称	类型	是否必需	描述
email	STRING	YES	
recvWindow	LONG	NO	
timestamp	LONG	YES	

杠杆逐仓账户划转 (MARGIN)

响应:

```
{
    //transaction id
    "tranId": 100000001
}
```

POST /sapi/v1/margin/isolated/transfer (HMAC SHA256)

权重(UID): 600

参数:

名称	类型	是否必需	描述
asset	STRING	YES	被划转的资产, 比如, BTC
symbol	STRING	YES	逐仓 symbol
transFrom	STRING	YES	"SPOT", "ISOLATED_MARGIN"
transTo	STRING	YES	"SPOT", "ISOLATED_MARGIN"
amount	DECIMAL	YES	划转数量
recvWindow	LONG	NO	赋值不能大于 60000
timestamp	LONG	YES	

获取杠杆逐仓划转历史 (USER_DATA)

响应:

```
{
  "rows": [
    {
      "amount": "0.10000000",
      "asset": "BNB",
      "status": "CONFIRMED",
      "timestamp": 1566898617000,
      "txId": 5240372201,
    }
  ]
}
```

```

    "transFrom": "SPOT",
    "transTo": "ISOLATED_MARGIN"
},
{
    "amount": "5.00000000",
    "asset": "USDT",
    "status": "CONFIRMED",
    "timestamp": 1566888436123,
    "txId": 5239810406,
    "transFrom": "ISOLATED_MARGIN",
    "transTo": "SPOT"
}
],
"total": 2
}

```

GET /sapi/v1/margin/isolated/transfer (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
asset	STRING	NO	
symbol	STRING	YES	逐仓 symbol
transFrom	STRING	NO	"SPOT", "ISOLATED_MARGIN"
transTo	STRING	NO	"SPOT", "ISOLATED_MARGIN"
startTime	LONG	NO	
endTime	LONG	NO	
current	LONG	NO	当前查询页。从 1开始。默认:1
size	LONG	NO	默认:10 最大:100
archived	STRING	NO	默认: false. 查询6个月以前的数据，需要设为 true
recvWindow	LONG	NO	赋值不能大于 60000
timestamp	LONG	YES	

- 查询时间范围最大不得超过30天。
- 若 startTime 和 endTime 没传，则默认返回最近7天数据
- 如果想查询6个月以前数据，设置 archived 为 true。

查询杠杆逐仓账户信息 (USER_DATA)

响应:

不传"symbols"的返回内容

```
{
  "assets": [
    {
      "baseAsset": {
        "asset": "BTC",
        "borrowEnabled": true,
        "borrowed": "0.00000000",
        "free": "0.00000000",
        "interest": "0.00000000",
        "locked": "0.00000000",
        "netAsset": "0.00000000",
        "netAssetOfBtc": "0.00000000",
        "repayEnabled": true,
        "totalAsset": "0.00000000"
      },
      "quoteAsset": {
        "asset": "USDT",
        "borrowEnabled": true,
        "borrowed": "0.00000000",
        "free": "0.00000000",
        "interest": "0.00000000",
        "locked": "0.00000000",
        "netAsset": "0.00000000",
        "netAssetOfBtc": "0.00000000",
        "repayEnabled": true,
        "totalAsset": "0.00000000"
      },
      "symbol": "BTCUSDT",
      "isolatedCreated": true,
      "enabled": true, // 账户是否启用, true-启用, false-停用
      "marginLevel": "0.00000000",
      "marginLevelStatus": "EXCESSIVE", // "EXCESSIVE", "NORMAL", "MARGIN_CALL", "PRE LIQUIDATION", "FORC",
      "marginRatio": "0.00000000",
      "indexPrice": "10000.00000000",
      "liquidatePrice": "1000.00000000",
      "liquidateRate": "1.00000000",
      "tradeEnabled": true
    }
  ],
  "totalAssetOfBtc": "0.00000000",
  "totalLiabilityOfBtc": "0.00000000",
  "totalNetAssetOfBtc": "0.00000000"
}

```

传"symbols"的返回内容

```
{
  "assets": [
    {
      "baseAsset": {
        "asset": "BTC",
        "borrowEnabled": true,
        "borrowed": "0.00000000",
        "free": "0.00000000",
        "interest": "0.00000000",
        "locked": "0.00000000",
        "netAsset": "0.00000000",
        "netAssetOfBtc": "0.00000000",
        "repayEnabled": true,
        "totalAsset": "0.00000000"
      }
    }
  ]
}
```

```

        "borrowed": "0.00000000",
        "free": "0.00000000",
        "interest": "0.00000000",
        "locked": "0.00000000",
        "netAsset": "0.00000000",
        "netAssetOfBtc": "0.00000000",
        "repayEnabled": true,
        "totalAsset": "0.00000000"
    },
    "quoteAsset": {
        "asset": "USDT",
        "borrowEnabled": true,
        "borrowed": "0.00000000",
        "free": "0.00000000",
        "interest": "0.00000000",
        "locked": "0.00000000",
        "netAsset": "0.00000000",
        "netAssetOfBtc": "0.00000000",
        "repayEnabled": true,
        "totalAsset": "0.00000000"
    },
    "symbol": "BTCUSDT",
    "isolatedCreated": true,
    "enabled": true, // 账户是否启用, true-启用, false-停用
    "marginLevel": "0.00000000",
    "marginLevelStatus": "EXCESSIVE", // "EXCESSIVE", "NORMAL", "MARGIN_CALL", "PRE LIQUIDATION", "FORC
    "marginRatio": "0.00000000",
    "indexPrice": "10000.00000000",
    "liquidatePrice": "1000.00000000",
    "liquidateRate": "1.00000000",
    "tradeEnabled": true
}
]
}

```

GET /sapi/v1/margin/isolated/account (HMAC SHA256)

权重(IP): 10

参数:

名称	类型	是否必需	描述
symbols	STRING	NO	最多可以传5个symbol; 由","分隔的字符串表示. e.g. "BTCUSDT,BNBUSDT,ADAUSDT"
recvWindow	LONG	NO	赋值不能大于 60000
timestamp	LONG	YES	

- 不传"symbols",返回所有杠杆逐仓资产
- 传"symbols", 将只会返回制定symbol的杠杆逐仓资产

杠杆逐仓账户停用 (TRADE)

响应

```
{
  "success": true,
  "symbol": "BTCUSDT"
}
```

`DELETE /sapi/v1/margin/isolated/account (HMAC SHA256)`

停用特定交易对的杠杆逐仓账户。每个交易对 24 小时内仅可停用一次。

权重(UID): 300

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
recvWindow	LONG	NO	不能大于 <code>60000</code>
timestamp	LONG	YES	

杠杆逐仓账户启用 (TRADE)

响应

```
{
  "success": true,
  "symbol": "BTCUSDT"
}
```

`POST /sapi/v1/margin/isolated/account (HMAC SHA256)`

启用特定交易对的杠杆逐仓账户（仅支持启用之前停用的账户）。

权重(UID): 300

参数

名称	类型	是否必需	描述
symbol	STRING	YES	
recvWindow	LONG	NO	不能大于 <code>60000</code>
timestamp	LONG	YES	

查询杠杆逐仓账户启用限制 (USER_DATA)

响应

```
{
  "enabledAccount": 5,
  "maxAccount": 20
}
```

GET /sapi/v1/margin/isolated/accountLimit (HMAC SHA256)

查询杠杆逐仓账户启用限制。

权重(IP): 1

参数:

名称	类型	是否必需	描述
recvWindow	LONG	NO	不能大于 60000
timestamp	LONG	YES	

查询逐仓杠杆交易对 (USER_DATA)

响应:

```
{
  "symbol": "BTCUSDT",
  "base": "BTC",
  "quote": "USDT",
  "isMarginTrade": true,
  "isBuyAllowed": true,
  "isSellAllowed": true
}
```

GET /sapi/v1/margin/isolated/pair (HMAC SHA256)

权重(IP): 10

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
recvWindow	LONG	NO	赋值不能大于 60000
timestamp	LONG	YES	

获取所有逐仓杠杆交易对(USER_DATA)

响应:

```
[
  {
    "base": "BNB",
    "isBuyAllowed": true,
    "isMarginTrade": true,
    "isSellAllowed": true,
    "quote": "BTC",
    "symbol": "BNBBTC"
  },
  {
    "base": "TRX",
    "isBuyAllowed": true,
    "isMarginTrade": true,
    "isSellAllowed": true,
    "quote": "BTC",
    "symbol": "TRXBTC"
  }
]
```

GET /sapi/v1/margin/isolated/allPairs (HMAC SHA256)

权重(IP): 10

参数:

名称	类型	是否必需	描述
recvWindow	LONG	NO	赋值不能大于 60000
timestamp	LONG	YES	

现货交易和杠杆利息BNB抵扣开关(USER_DATA)

响应:

```
{
  "spotBNBBurn": true,
  "interestBNBBurn": false
}
```

POST /sapi/v1/bnbBurn (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
spotBNBBurn	STRING	NO	"true" or "false", 是否使用BNB支付现货交易的手续费
interestBNBBurn	STRING	NO	"true" or "false", 是否使用BNB支付杠杆贷款的利息
recvWindow	LONG	NO	赋值不能大于 60000

名称	类型	是否必需	描述
timestamp	LONG	YES	

- "spotBNBBurn" 和 "interestBNBBurn" 二者必须传至少一个

获取BNB抵扣开关状态 (USER_DATA)

响应:

```
{
  "spotBNBBurn": true,
  "interestBNBBurn": false
}
```

GET /sapi/v1/bnbBurn (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
recvWindow	LONG	NO	赋值不能大于 60000
timestamp	LONG	YES	

获取杠杆利率历史 (USER_DATA)

响应:

```
[
  {
    "asset": "BTC",
    "dailyInterestRate": "0.00025000",
    "timestamp": 1611544731000,
    "vipLevel": 1
  },
  {
    "asset": "BTC",
    "dailyInterestRate": "0.00035000",
    "timestamp": 1610248118000,
    "vipLevel": 1
  }
]
```

GET /sapi/v1/margin/interestRateHistory (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
asset	STRING	YES	
vipLevel	INT	NO	默认用户当前等级
startTime	LONG	NO	默认7天前
endTime	LONG	NO	默认当天，时间间隔最大为1个月
recvWindow	LONG	NO	赋值不能大于 60000
timestamp	LONG	YES	

获取全仓杠杆利率及限额 (USER_DATA)

响应:

```
[
  {
    "vipLevel": 0,
    "coin": "BTC",
    "transferIn": true,
    "borrowable": true,
    "dailyInterest": "0.00026125",
    "yearlyInterest": "0.0953",
    "borrowLimit": "180",
    "marginablePairs": [
      "BNBBTC",
      "TRXBTC",
      "ETHBTC",
      "BTCUSDT"
    ]
  }
]
```

GET /sapi/v1/margin/crossMarginData (HMAC SHA256)

通过VIP等级或用户当前VIP等级获取全仓杠杆利率及限额，如：<https://www.binance.com/en/margin-fee>

权重: 1 指定币种; 5 币种参数缺失

参数(IP):

名称	类型	是否必需	描述
vipLevel	INT	NO	默认为用户当前VIP等级
coin	STRING	NO	
recvWindow	LONG	NO	赋值不能大于 60000
timestamp	LONG	YES	

获取逐仓杠杆利率及限额 (USER_DATA)

响应:

```
[
  {
    "vipLevel": 0,
    "symbol": "BTCUSDT",
    "leverage": "10",
    "data": [
      {
        "coin": "BTC",
        "dailyInterest": "0.00026125",
        "borrowLimit": "270"
      },
      {
        "coin": "USDT",
        "dailyInterest": "0.000475",
        "borrowLimit": "2100000"
      }
    ]
  }
]
```

GET /sapi/v1/margin/isolatedMarginData (HMAC SHA256)

通过VIP等级或用户当前VIP等级获取逐仓杠杆利率及限额，如：<https://www.binance.com/en/margin-fee>

权重(IP): 1 指定交易对; 10 交易对参数缺失

参数:

名称	类型	是否必须	描述
vipLevel	INT	NO	默认为用户当前VIP等级
symbol	STRING	NO	
recvWindow	LONG	NO	赋值不能大于 60000
timestamp	LONG	YES	

获取逐仓档位信息 (USER_DATA)

响应:

```
[
  {
    "symbol": "BTCUSDT",
    "tier": 1,
    "effectiveMultiple": "10",
    "initialRiskRatio": "1.111",
    "liquidationRiskRatio": "1.05",
  }
]
```

```

    "baseAssetMaxBorrowable": "9",
    "quoteAssetMaxBorrowable": "70000"
}
]

```

GET /sapi/v1/margin/isolatedMarginTier (HMAC SHA256)

通过档位获取逐仓杠杆档位数据，如：<https://www.binance.com/en/margin-data>

权重(IP): 1

参数:

名称	类型	是否必须	描述
symbol	STRING	YES	
tier	INTEGER	NO	不传则返回所有逐仓杠杆档位
recvWindow	LONG	NO	赋值不能大于 60000
timestamp	LONG	YES	

查询目前杠杆账户下单数 (TRADE)

响应:

```
[
  {
    "rateLimitType": "ORDERS",
    "interval": "SECOND",
    "intervalNum": 10,
    "limit": 10000,
    "count": 0
  },
  {
    "rateLimitType": "ORDERS",
    "interval": "DAY",
    "intervalNum": 1,
    "limit": 20000,
    "count": 0
  }
]
```

GET /sapi/v1/margin/rateLimit/order

获取用户在当前时间区间内的杠杆账户下单总数。

权重(IP): 20

参数:

名称	类型	是否必需	描述

名称	类型	是否必需	描述
isIsolated	STRING	NO	是否逐仓杠杆 , "TRUE", "FALSE", 默认 "FALSE"
symbol	STRING	NO	逐仓交易对 , 查询逐仓杠杆账户必需
recvWindow	LONG	NO	赋值不能大于 60000
timestamp	LONG	YES	

杠杆小额资产转换BNB历史 (USER_DATA)

响应:

```
{
  "total": 8, //共计发生过的转换笔数
  "userAssetDribbles": [
    {
      "operateTime": 1615985535000,
      "totalTransferredAmount": "0.00132256", //本次转换所得BNB
      "totalServiceChargeAmount": "0.00002699", //本次转换手续费(BNB)
      "transId": 45178372831,
      "userAssetDribbleDetails": [ //本次转换的细节
        {
          "transId": 4359321,
          "serviceChargeAmount": "0.000009",
          "amount": "0.0009",
          "operateTime": 1615985535000,
          "transferredAmount": "0.000441",
          "fromAsset": "USDT"
        },
        {
          "transId": 4359321,
          "serviceChargeAmount": "0.00001799",
          "amount": "0.0009",
          "operateTime": 1615985535000,
          "transferredAmount": "0.00088156",
          "fromAsset": "ETH"
        }
      ]
    },
    {
      "operateTime": 1616203180000,
      "totalTransferredAmount": "0.00058795",
      "totalServiceChargeAmount": "0.000012",
      "transId": 4357015,
      "userAssetDribbleDetails": [
        {
          "transId": 4357015,
          "serviceChargeAmount": "0.00001",
          "amount": "0.001",
          "operateTime": 1616203180000,
          "transferredAmount": "0.00049",
          "fromAsset": "USDT"
        }
      ]
    }
  ]
}
```

```

        },
        {
            "transId": 4357015,
            "serviceChargeAmount": "0.000002",
            "amount": "0.0001",
            "operateTime": 1616203180000,
            "transferredAmount": "0.00009795",
            "fromAsset": "ETH"
        }
    ]
}
]
}
}

```

GET /sapi/v1/margin/dribblet (HMAC SHA256)

查询用户杠杆账户小额资产转换BNB历史信息

参数:

名称	类型	是否必需	描述
startTime	LONG	NO	
endTime	LONG	NO	
recvWindow	LONG	NO	
timestamp	LONG	YES	

WebSocket账户信息推送

- 本篇所列出API接口的base url : <https://api.binance.com>
- 用于订阅账户数据的 `listenKey` 从创建时刻起有效期为60分钟
- 可以通过 `PUT` 一个 `listenKey` 延长60分钟有效期
- 可以通过 `DELETE` 一个 `listenKey` 立即关闭当前数据流，并使该 `listenKey` 无效
- 在具有有效 `listenKey` 的帐户上执行 `POST` 将返回当前有效的 `listenKey` 并将其有效期延长60分钟
- websocket接口的baseurl: <wss://stream.binance.com:9443>
- U订阅账户数据流的stream名称为 `/ws/<listenKey>` 或 `/stream?streams=<listenKey>`
- 每个链接有效期不超过24小时，请妥善处理断线重连。
- 账户数据流的消息不保证严格时间序, 请使用 `E` 字段进行排序

Listen Key(现货账户)

生成 LISTEN KEY (USER_STREAM)

响应

```
{
    "listenKey": "pqia91ma19a5s61cv6a81va65sdf19v8a65a1a5s61cv6a81va65sdf19v8a65a1"
}
```

}

`POST /api/v3/userDataStream`

开始一个新的数据流。除非发送 keepalive，否则数据流于60分钟后关闭。如果该帐户具有有效的 `listenKey`，则将返回该 `listenKey` 并将其有效期延长60分钟。 权重: 1

参数:

NONE

数据源: 缓存**延长 LISTEN KEY 有效期 (USER_STREAM)****响应**

{}

`PUT /api/v3/userDataStream`

有效期延长至本次调用后60分钟,建议每30分钟发送一个 ping。

权重: 1**参数:**

名称	类型	是否必需	描述
listenKey	STRING	YES	

数据源: 缓存**关闭 LISTEN KEY (USER_STREAM)****响应**

{}

`DELETE /api/v3/userDataStream`

关闭用户数据流。

权重: 1**参数:**

名称	类型	是否必需	描述
listenKey	STRING	YES	

数据源: 缓存

Listen Key(杠杆账户)

生成 LISTEN KEY (USER_STREAM)

响应

```
{"listenKey": "T3ee22BIYuWqmVne0HNq2A2WsF1EtLhvWCTItw6ffhdmjifQ2tRbuKkTHhr"}
```

`POST /sapi/v1/userDataStream`

权重: 1

参数:

NONE

延长 LISTEN KEY 有效期 (USER_STREAM)**响应**

```
{}
```

`PUT /sapi/v1/userDataStream`

权重: 1

参数:

名称	类型	是否必需	描述
listenKey	STRING	YES	

关闭 LISTENKEY (USER_STREAM)**响应**

```
{}
```

`DELETE /sapi/v1/userDataStream`

权重: 1

参数:

名称	类型	是否必需	描述
listenKey	STRING	YES	

Listen Key(逐仓杠杆账户)

生成 LISTEN KEY (USER_STREAM)**响应:**

```
{
  "listenKey": "T3ee22BIYuWqmVne0HNq2A2WsF1EtLhvWCTItw6ffhdmjifQ2tRbuKkTHhr"
}
```

POST /sapi/v1/userDataStream/isolated

权重: 1

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	

延长 LISTEN KEY 有效期 (USER_STREAM)

响应:

{}

PUT /sapi/v1/userDataStream/isolated

权重: 1

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
listenKey	STRING	YES	

关闭 LISTENKEY (USER_STREAM)

响应:

{ }

```
DELETE /sapi/v1/userDataStream/isolated
```

权重: 1

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	
listenKey	STRING	YES	

Payload: 账户更新

每当帐户余额发生更改时，都会发送一个事件 `outboundAccountPosition`，其中包含可能由生成余额变动的事件而变动的资产。

Payload

```
{  
  "e": "outboundAccountPosition", // 事件类型  
  "E": 1564034571105, // 事件时间
```

```

"u": 1564034571073,           // 账户末次更新时间戳
"B": [
  {
    "a": "ETH",                // 资产名称
    "f": "10000.000000",       // 可用余额
    "l": "0.000000"           // 冻结余额
  }
]
}

```

Payload: 余额更新

Payload

```

{
  "e": "balanceUpdate",        //Event Type
  "E": 1573200697110,         //Event Time
  "a": "ABC",                 //Asset
  "d": "100.00000000",        //Balance Delta
  "T": 1573200697068         //Clear Time
}

```

当下列情形发生时更新:

- 账户发生充值或提取
- 交易账户之间发生划转(例如 现货向杠杆账户划转)

Payload: 订单更新

订单通过executionReport事件进行更新。

请查阅文档公开API参数以及以下文档，以获取相关的枚举定义。

通过将 Δ 除以 Δ 可以找到平均价格。

Payload

```

{
  "e": "executionReport",      // 事件类型
  "E": 1499405658658,         // 事件时间
  "s": "ETHBTC",              // 交易对
  "c": "mUvoqJxFIILMdfAW5iGSOW", // clientOrderId
  "S": "BUY",                 // 订单方向
  "o": "LIMIT",               // 订单类型
  "f": "GTC",                 // 有效方式
  "q": "1.00000000",          // 订单原始数量
  "p": "0.10264410",          // 订单原始价格
  "P": "0.00000000",          // 止盈止损单触发价格
  "d": 4,                     // 追踪止损(Trailing Delta) 只有在追踪止损订单中才会推送。
  "F": "0.00000000",          // 冰山订单数量
  "g": -1,                    // OCO订单 OrderListId
}

```

```

    "C": "",                                // 原始订单自定义ID(原始订单，指撤单操作的对象。撤单本身被视为另一个订单)
    "x": "NEW",                             // 本次事件的具体执行类型
    "X": "NEW",                             // 订单的当前状态
    "r": "NONE",                           // 订单被拒绝的原因
    "i": 4293153,                          // orderId
    "l": "0.00000000",                     // 订单末次成交量
    "z": "0.00000000",                     // 订单累计已成交量
    "L": "0.00000000",                     // 订单末次成交价格
    "n": "0",                               // 手续费数量
    "N": null,                            // 手续费资产类别
    "T": 1499405658657,                   // 成交时间
    "t": -1,                               // 成交ID
    "I": 8641984,                          // 请忽略
    "w": true,                            // 订单是否在订单簿上?
    "m": false,                            // 该成交是作为挂单成交吗?
    "M": false,                            // 请忽略
    "O": 1499405658657,                   // 订单创建时间
    "Z": "0.00000000",                     // 订单累计已成交金额
    "Y": "0.00000000",                     // 订单末次成交金额
    "Q": "0.00000000",                     // Quote Order Qty
    "j": 1,                                // 策略ID; 只有下单的时候提供了strategyId才会显示
    "J": 1000000                          // 策略类型; 只有下单的时候提供了strategyType才会显示
}

```

执行类型:

- NEW 新订单
- CANCELED 订单被取消
- REPLACED (保留字段, 当前未使用)
- REJECTED 新订单被拒绝 (这信息只会在撤消挂单再下单中发生, 下新订单被拒绝但撤消挂单请求成功。)
- TRADE 订单有新成交
- EXPIRED 订单失效(根据订单的Time In Force参数)

如果订单是OCO，则除了显示“executionReport”事件外，还将显示一个名为“listStatus”的事件。

Payload

```
{
  "e": "listStatus",                      // 事件类型
  "E": 1564035303637,                    // 事件时间
  "s": "ETHBTC",                         // 交易对
  "g": 2,                                // OrderListId
  "c": "OCO",                            // Contingency Type
  "l": "EXEC_STARTED",                   // List Status Type
  "L": "EXECUTING",                      // List Order Status
  "r": "NONE",                           // List 被拒绝的原因
  "C": "F4QN4G8D1FATF1IUQ0cjddD",       // List Client Order ID
  "T": 1564035303625,                   // 成交时间
  "O": [
    {
      "s": "ETHBTC",                      // 交易对
      "i": 17,                            // orderId
      "c": "AJYsMjErWJesZvqlJCTUgL" // clientOrderId
    },
    {
      "s": "ETHBTC",                      // 交易对
      "i": 18,                            // orderId
      "c": "bfYPSQdLoqAJeNrOr9adzq"
    }
  ]
}
```

```
    }
]
}
```

币安宝接口

- 这些接口用于币安宝产品。更多细节, 请参考币安宝页面。

获取活期产品列表 (USER_DATA)

响应:

```
[
{
  "asset": "BTC",
  "avgAnnualInterestRate": "0.05000000"
  "tierAnnualInterestRate": {
    "0-5BTC": 0.05,
    "5-10BTC": 0.03,
    ">10BTC": 0.01
  },
  "canPurchase": true,
  "canRedeem": true,
  "dailyInterestPerThousand": "0.00685000", //弃用
  "featured": true,
  "minPurchaseAmount": "0.01000000",
  "productId": "BTC001",
  "purchasedAmount": "16.32467016",
  "status": "PURCHASING", //PREHEATING: 预热; PURCHASING: 申购中; END: 已结束
  "upLimit": "200.00000000",
  "upLimitPerUser": "5.00000000"
},
{
  "asset": "BUSD",
  "avgAnnualInterestRate": "0.01228590",
  "tierAnnualInterestRate": "",
  "canPurchase": true,
  "canRedeem": true,
  "dailyInterestPerThousand": "0.03836000", //弃用
  "featured": true,
  "minPurchaseAmount": "0.10000000",
  "productId": "BUSD001",
  "purchasedAmount": "10.38932339",
  "status": "PURCHASING", //PREHEATING: 预热; PURCHASING: 申购中; END: 已结束
  "upLimit": "100000.00000000",
  "upLimitPerUser": "50000.00000000"
}
]
```

```
GET /sapi/v1/lending/daily/product/list (HMAC SHA256)
```

权重(IP): 1

参数:

名称	类型	是否必需	描述
status	ENUM	NO	"ALL", "SUBSCRIBABLE", "UNSUBSCRIBABLE"; 默认: "ALL"
featured	STRING	NO	"ALL", "TRUE"; 默认: "ALL"
current	LONG	NO	当前页面. 默认: 1, 最小: 1
size	LONG	NO	默认: 50, 最大: 100
recvWindow	LONG	NO	不能大于 60000
timestamp	LONG	YES	

获取用户当日剩余活期可申购余额 (USER_DATA)

响应:

```
{
  "asset": "BUSD",
  "leftQuota": "50000.00000000"
}
```

```
GET /sapi/v1/lending/daily/userLeftQuota (HMAC SHA256)
```

权重(IP): 1

参数:

名称	类型	是否必需	描述
productId	STRING	YES	
recvWindow	LONG	NO	不能大于 60000
timestamp	LONG	YES	

申购活期产品 (USER_DATA)

响应:

```
{
  "purchaseId": 40607
}
```

`POST /sapi/v1/lending/daily/purchase (HMAC SHA256)`

权重(IP): 1

频次限制 : 每个账户最多三秒一次

参数:

名称	类型	是否必需	描述
productId	STRING	YES	
amount	DECIMAL	YES	
recvWindow	LONG	NO	不能大于 60000
timestamp	LONG	YES	

- 您需要为API Key开通允许现货和杠杆交易权限才能发送此请求

获取用户当日活期可赎回余额 (USER_DATA)

响应:

```
{
  "asset": "USDT",
  "dailyQuota": "10000000.00000000",
  "leftQuota": "0.00000000",
  "minRedemptionAmount": "0.10000000"
}
```

`GET /sapi/v1/lending/daily/userRedemptionQuota (HMAC SHA256)`

权重(IP): 1

参数:

名称	类型	是否必需	描述
productId	STRING	YES	
type	ENUM	YES	"FAST", "NORMAL"
recvWindow	LONG	NO	不能大于 60000
timestamp	LONG	YES	

赎回活期产品 (USER_DATA)

响应:

```
[]
```

`POST /sapi/v1/lending/daily/redeem (HMAC SHA256)`

权重(IP): 1

参数:

名称	类型	是否必需	描述
productId	STRING	YES	
amount	DECIMAL	YES	
type	ENUM	YES	"FAST", "NORMAL"
recvWindow	LONG	NO	不能大于 60000
timestamp	LONG	YES	

- 您需要为API Key开通允许现货和杠杆交易权限才能发送此请求

用户活期产品持仓 (USER_DATA)

响应:

```
[
  {
    "tierAnnualInterestRate": {
      "0-5BTC": 0.05,
      "5-10BTC": 0.03,
      ">10BTC": 0.01
    },
    "annualInterestRate": "0.02599895",
    "asset": "USDT",
    "avgAnnualInterestRate": "0.02599895",
    "canRedeem": true,
    "dailyInterestRate": "0.00007123",
    "freeAmount": "75.46000000",
    "freezeAmount": "0.00000000", // abandoned
    "lockedAmount": "0.00000000", // abandoned
    "productId": "USDT001",
    "productName": "USDT",
    "redeemingAmount": "0.00000000",
    "todayPurchasedAmount": "0.00000000",
    "totalAmount": "75.46000000",
    "totalInterest": "0.22759183"
  }
]
```

`GET /sapi/v1/lending/daily/token/position (HMAC SHA256)`

权重(IP): 1

参数:

名称	类型	是否必需	描述
asset	STRING	NO	
recvWindow	LONG	NO	不能大于 60000
timestamp	LONG	YES	

查询定期/活动产品列表 (USER_DATA)

响应:

```
[
  {
    "asset": "USDT",
    "displayPriority": 1,
    "duration": 90,
    "interestPerLot": "1.35810000",
    "interestRate": "0.05510000",
    "lotSize": "100.00000000",
    "lotsLowLimit": 1,
    "lotsPurchased": 74155,
    "lotsUpLimit": 80000,
    "maxLotsPerUser": 2000,
    "needKyc": False,
    "projectId": "CUSDT90DAYSS001",
    "projectName": "USDT",
    "status": "PURCHASING",
    "type": "CUSTOMIZED_FIXED",
    "withAreaLimitation": False
  }
]
```

GET /sapi/v1/lending/project/list (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
asset	STRING	NO	
type	ENUM	YES	"CUSTOMIZED_FIXED"定期 , "ACTIVITY" 活动
status	ENUM	NO	"ALL", "SUBSCRIBABLE", "UNSUBSCRIBABLE"; 默认 "ALL"
isSortAsc	BOOLEAN	NO	默认 "true"
sortBy	ENUM	NO	"START_TIME", "LOT_SIZE", "INTEREST_RATE", "DURATION"; 默认 "START_TIME"
current	LONG	NO	分页页码. 默认:1
size	LONG	NO	单页显示条数 , 默认:10 最大:100

名称	类型	是否必需	描述
recvWindow	LONG	NO	不能大于 60000
timestamp	LONG	YES	

申购定期/活动产品 (USER_DATA)

响应:

```
{
    purchaseId: "18356"
}
```

POST /sapi/v1/lending/customizedFixed/purchase (HMAC SHA256)

权重(IP): 1

频次限制 : 每个账户最多三秒一次

参数:

名称	类型	是否必需	描述
projectId	STRING	YES	
lot	LONG	YES	申购手数
recvWindow	LONG	NO	不能大于 60000
timestamp	LONG	YES	

- 您需要为API Key开通允许现货和杠杆交易权限才能发送此请求

用户定期/活动持仓 (USER_DATA)

响应:

```
[
{
    "asset": "USDT",
    "canTransfer": True,
    "createTimestamp": 1587010770000,
    "duration": 14,
    "endTime": 1588291200000,
    "interest": "0.19950000",
    "interestRate": "0.05201250",
    "lot": 1,
    "positionId": 51724,
    "principal": "100.00000000",
    "projectId": "CUSDT14DAYSS001",
    "status": "Normal"
}
```

```

    "productName": "USDT",
    "purchaseTime": 1587010771000,
    "redeemDate": "2020-05-01",
    "startTime": 1587081600000,
    "status": "HOLDING",
    "type": "CUSTOMIZED_FIXED"
}
]

```

GET /sapi/v1/lending/project/position/list (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
asset	STRING	NO	
projectId	STRING	NO	
status	ENUM	NO	"HOLDING", "REDEEMED"
recvWindow	LONG	NO	不能大于 60000
timestamp	LONG	YES	

币安宝账户信息 (USER_DATA)

响应:

```

{
  "positionAmountVos": [
    {
      "amount": "75.46000000",
      "amountInBTC": "0.01044819",
      "amountInUSDT": "75.46000000",
      "asset": "USDT"
    },
    {
      "amount": "1.67072036",
      "amountInBTC": "0.00023163",
      "amountInUSDT": "1.67289230",
      "asset": "BUSD"
    }
  ],
  "totalAmountInBTC": "0.01067982",
  "totalAmountInUSDT": "77.13289230",
  "totalFixedAmountInBTC": "0.00000000",
  "totalFixedAmountInUSDT": "0.00000000",
  "totalFlexibleInBTC": "0.01067982",
  "totalFlexibleInUSDT": "77.13289230"
}

```

GET /sapi/v1/lending/union/account (HMAC SHA256)

权重(IP): 1**参数:**

名称	类型	是否必需	描述
recvWindow	LONG	NO	不能大于 60000
timestamp	LONG	YES	

获取申购记录 (USER_DATA)

响应:

活期产品

```
[
  {
    "amount": "100.00000000",
    "asset": "USDT",
    "createTime": 1575018510000,
    "lendingType": "DAILY",
    "productName": "USDT",
    "purchaseId": 26055,
    "status": "SUCCESS"
  }
]
```

定期/活动产品

```
[
  {
    "amount": "100.00000000",
    "asset": "USDT",
    "createTime": 1575018453000,
    "lendingType": "ACTIVITY",
    "lot": 1,
    "productName": "【Special】USDT 7D (8%)",
    "purchaseId": 36857,
    "status": "SUCCESS"
  },
  {
    "amount": "100.00000000",
    "asset": "USDT",
    "createTime": 1587010770000,
    "lendingType": "CUSTOMIZED_FIXED",
    "lot": 1,
    "productName": "USDT",
    "purchaseId": 55841,
    "status": 'SUCCESS'
  }
]
```

`GET /sapi/v1/lending/union/purchaseRecord (HMAC SHA256)`

权重(IP): 1

参数:

名称	类型	是否必需	描述
lendingType	ENUM	YES	"DAILY" 表示活期, "ACTIVITI" 表示活动, "CUSTOMIZED_FIXED" 表示定期
asset	STRING	NO	
startTime	LONG	NO	
endTime	LONG	NO	
current	LONG	NO	Currently querying page. Start from 1. Default:1
size	LONG	NO	Default:10, Max:100
recvWindow	LONG	NO	不能大于 60000
timestamp	LONG	YES	

- `startTime` 和 `endTime` 的最大间隔为 30 天
- 若 `startTime` 和 `endTime` 均未发送，则默认返回最近 30 天记录

获取赎回记录 (USER_DATA)

响应:

活期产品

```
[  
  {  
    "amount": "10.54000000",  
    "asset": "USDT",  
    "createTime": 1577257222000,  
    "principal": "10.54000000",  
    "projectId": "USDT001",  
    "projectName": "USDT",  
    "status": "PAID",  
    "type": "FAST"  
  }  
]
```

定期/活动产品

```
[  
  {  
    "amount": "0.07070000",  
    "asset": "USDT",  
    "createTime": 1566200161000,  
    "interest": "0.00070000",  
    "principal": "0.07070000",  
    "projectId": "USDT001",  
    "projectName": "USDT",  
    "status": "PAID",  
    "type": "FAST"  
  }  
]
```

```

        "principal": "0.07000000",
        "projectId": "test06",
        "productName": "USDT 1 day (10% annualized)",
        "startTime": 1566198000000,
        "status": "PAID"
    }
]

```

`GET /sapi/v1/lending/union/redemptionRecord (HMAC SHA256)`

权重(IP): 1

参数:

名称	类型	是否必需	描述
lendingType	ENUM	YES	"DAILY" 表示活期, "ACTIVITI" 表示活动, "CUSTOMIZED_FIXED" 表示定期
asset	STRING	NO	
startTime	LONG	NO	
endTime	LONG	NO	
current	LONG	NO	Currently querying page. Start from 1. Default:1
size	LONG	NO	Default:10, Max:100
recvWindow	LONG	NO	不能大于 60000
timestamp	LONG	YES	

- `startTime` 和 `endTime` 的最大间隔为 30 天
- 若 `startTime` 和 `endTime` 均未发送，则默认返回最近 30 天记录

获取利息历史 (USER_DATA)

响应:

```

[
  {
    "asset": "BUSD",
    "interest": "0.00006408",
    "lendingType": "DAILY",
    "productName": "BUSD",
    "time": 1577233578000
  },
  {
    "asset": "USDT",
    "interest": "0.00687654",
    "lendingType": "DAILY",
    "productName": "USDT",
    "time": 1577233562000
  }
]

```

GET /sapi/v1/lending/union/interestHistory (HMAC SHA256)

权重(IP): 1**参数:**

名称	类型	是否必需	描述
lendingType	ENUM	YES	"DAILY" 表示活期, "ACTIVITY" 表示活动, "CUSTOMIZED_FIXED" 表示定期
asset	STRING	NO	
startTime	LONG	NO	
endTime	LONG	NO	
current	LONG	NO	Currently querying page. Start from 1. Default:1
size	LONG	NO	Default:10, Max:100
recvWindow	LONG	NO	不能大于 60000
timestamp	LONG	YES	

- `startTime` 和 `endTime` 的最大间隔为30天
- 若 `startTime` 和 `endTime` 均未发送，则默认返回最近30天记录

定期/活动持仓转活期持仓 (USER_DATA)

响应:

```
{
  "dailyPurchaseId": 862290,
  "success": true,
  "time": 1577233578000
},
```

POST /sapi/v1/lending/positionChanged (HMAC SHA256)

权重(IP): 1**参数:**

名称	类型	是否必需	描述
projectId	STRING	YES	
lot	LONG	YES	
positionId	LONG	NO	
recvWindow	LONG	NO	不能大于 60000
timestamp	LONG	YES	

- 定期持仓转活期持仓，需必填positionId
- 您需要为API Key开通允许现货和杠杆交易权限才能发送此请求

Staking 接口

以下接口适用于币安Staking产品。更多细节, 请参考币安Staking页面

查询Staking产品列表(USER_DATA)

响应:

```
[
  {
    "projectId": "Axs*90",
    "detail": {
      "asset": "AXS", //锁仓资产
      "rewardAsset": "AXS", //收益资产
      "duration": 90, //锁仓周期(天)
      "renewable": true, //项目支持续期锁仓
      "apy": "1.2069"
    },
    "quota": {
      "totalPersonalQuota": "2", //个人总额度
      "minimum": "0.001" //单笔最小
    }
  },
  {
    "projectId": "Fio*90",
    "detail": {
      "asset": "FIO",
      "rewardAsset": "FIO",
      "duration": 90,
      "renewable": true,
      "apy": "1.0769"
    },
    "quota": {
      "totalPersonalQuota": "600",
      "minimum": "0.1"
    }
  }
]
```

GET /sapi/v1/staking/productList (HMAC SHA256)

获取可锁仓的Staking产品

权重(IP): 1

参数:

名称	类型	是否必需	描述
product	ENUM	YES	"STAKING" 是Staking, "F_DEFI" 是DeFi活期挖矿, "L_DEFI" 是DeFi定期挖矿
asset	STRING	NO	
current	LONG	NO	当前查询页。开始值 1，默认:1
size	LONG	NO	默认 : 10 , 最大 : 100
recvWindow	LONG	NO	
timestamp	LONG	YES	

申购锁仓产品(USER_DATA)

响应:

```
{
  "positionId": "12345",
  "success": true
}
```

POST /sapi/v1/staking/purchase (HMAC SHA256)

权重(IP): 1

频次限制：每个账户最多三秒一次

参数:

名称	类型	是否必需	描述
product	ENUM	YES	"STAKING" 是Staking, "F_DEFI" 是DeFi活期挖矿, "L_DEFI" 是DeFi定期挖矿
productId	STRING	YES	
amount	DECIMAL	YES	
renewable	STRING	NO	true或者false，默认为false。仅在产品为 "STAKING" 或 "L_DEFI"时生效
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 您需要为API Key开通允许现货和杠杆交易权限才能发送此请求

赎回锁仓产品(USER_DATA)

响应:

```
{
  "success":true
}
```

POST /sapi/v1/staking/redeem (HMAC SHA256)

赎回锁仓产品。Staking和DeFi定期挖矿的赎回类型属于提前赎回，提前赎回将扣减已获得的收益。

权重(IP): 1

参数:

名称	类型	是否必需	描述
product	ENUM	YES	"STAKING" 是Staking, "F_DEFIL" 是DeFi活期挖矿, "L_DEFIL" 是DeFi定期挖矿
positionId	STRING	NO	"1234", 当产品为"STAKING" 或 "L_DEFIL"时为必填项
productId	STRING	YES	
amount	DECIMAL	NO	当产品为"F_DEFIL" 时为必填项
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 您需要为API Key开通允许现货和杠杆交易权限才能发送此请求

查看个人持仓(USER_DATA)

响应:

```
[
  {
    "positionId": "123123", //Staking持仓的编号
    "projectId": "AxS*90", //Staking项目的编码
    "asset": "AXS", //已锁仓资产
    "amount": "122.09202928", //已锁仓数量
    "purchaseTime": "1646182276000", //申购时间
    "duration": "60", //锁仓时间(天)
    "accrualDays": "4", //计息天数
    "rewardAsset": "AXS", //收益资产
    "APY": "0.2032",
    "rewardAmt": "5.17181528", //收益数量
    "extraRewardAsset": "BNB", //额外锁仓类型的收益资产
    "extraRewardAPY": "0.0203", //额外锁仓类型的年化收益率
    "estExtraRewardAmt": "5.17181528", //额外锁仓类型的收益，订单到期时分发
    "nextInterestPay": "1.29295383", //下次预估发放利息数量
    "nextInterestPayDate": "1646697600000", //下次发息日期
    "payInterestPeriod": "1", //发息周期
    "redeemAmountEarly": "2802.24068892", //提前赎回数量
    "interestEndDate": "1651449600000", //计息结束日
    "deliverDate": "1651536000000", //赎回到账时间
  }
]
```

```

    "redeemPeriod": "1",          //赎回时间间隔
    "redeemingAmt": "232.2323",   //赎回中的数量
    "partialAmtDeliverDate": "1651536000000", //订单部分赎回的到账时间
    "canRedeemEarly": true,      //为true时可以操作提前赎回
    "renewable": true,           //为true时可以操作自动续期
    "type": "AUTO",             //订单类型为自动续期或是普通订单
    "status": "HOLDING"         //状态
}
]

```

GET /sapi/v1/staking/position (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
product	ENUM	YES	"STAKING" 是Staking, "F_DEFI" 是DeFi活期挖矿, "L_DEFI" 是DeFi定期挖矿
productId	STRING	NO	
asset	STRING	NO	
current	LONG	NO	当前查询页。开始值 1， 默认:1
size	LONG	NO	默认 : 10 , 最大 : 100
recvWindow	LONG	NO	
timestamp	LONG	YES	

查看Staking历史记录(USER_DATA)

响应:

```

[
{
    "positionId": "123123",
    "time": 1575018510000,
    "asset": "BNB",
    "project": "BSC",      //DeFi挖矿的项目方
    "amount": "21312.23223",
    "lockPeriod": "30",
    "deliverDate": "1575018510000", //赎回到账时间
    "type": "AUTO",        //仅申购记录展示
    "status": "success"
}
]

```

GET /sapi/v1/staking/stakingRecord (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
product	ENUM	YES	"STAKING" 是Staking, "F_DEFIL" 是DeFi活期挖矿, "L_DEFIL" 是DeFi定期挖矿
txType	ENUM	YES	申购："SUBSCRIPTION", 赎回："REDEMPTION", 利息："INTEREST"
asset	STRING	NO	
startTime	LONG	NO	
endTime	LONG	NO	
current	LONG	NO	当前查询页。开始值 1, 默认:1
size	LONG	NO	默认：10，最大：100
recvWindow	LONG	NO	
timestamp	LONG	YES	

- `startTime`和`endTime`的最大间隔为3个月
- 若`startTime`和`endTime`均未发送，则默认返回最近30天记录

设置自动续期(USER_DATA)

响应:

```
{
  "success":true
}
```

POST /sapi/v1/staking/setAutoStaking (HMAC SHA256)

设置Staking和DeFi定期挖矿的自动续期功能

权重(IP): 1

参数:

名称	类型	是否必需	描述
product	ENUM	YES	"STAKING" 是Staking, "L_DEFIL" 是DeFi定期挖矿
positionId	STRING	YES	
renewable	STRING	YES	true 或者 false
recvWindow	LONG	NO	
timestamp	LONG	YES	

查询Staking个人剩余额度(USER_DATA)

响应:

```
[  
 {  
   "leftPersonalQuota": "1000" // 用户剩余可用额度  
 }  
]
```

GET /sapi/v1/staking/personalLeftQuota(HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
product	ENUM	YES	"STAKING" 是Staking, "F_DEFI" 是DeFi活期挖矿, "L_DEFI" 是DeFi定期挖矿
productId	STRING	YES	
recvWindow	LONG	NO	
timestamp	LONG	YES	

矿池接口

- 这些接口作用于币安矿池。更多细节, 参考币安矿池页面。

获取算法(MARKET_DATA)

响应:

```
{  
  "code": 0,  
  "msg": "",  
  "data": [  
    {  
      "algoName": "sha256",  
      "algoId": 1,  
      "poolIndex": 0,  
      "unit": "h/s"  
    }  
  ]  
}
```

`GET /sapi/v1/mining/pub/algoList (HMAC SHA256)`

权重(IP): 1

参数:

None

获取币种(MARKET_DATA)

响应:

```
{
  "code": 0,
  "msg": "",
  "data": [
    {
      "coinName": "BTC", // 币种名称
      "coinId": 1, // id
      "poolIndex": 0, // 排序
      "algoId": 1, // 所属算法
      "algoName": "sha256" //所属算法名称
    }
  ]
}
```

`GET /sapi/v1/mining/pub/coinList (HMAC SHA256)`

权重(IP): 1

参数:

None

请求矿工列表明细 (USER_DATA)

响应:

```
{
  "code": 0,
  "msg": "",
  "data": [
    {
      "workerName": "bhdc1.16A10404B", //挖矿用户名
      "type": "H_hashrate", // 小时算力类型
      "hashrateDatas": [
        {
          "time": 1587902400000, // 时间
          "hashrate": "0", // 算力
          "reject": 0 //拒绝率
        },
        {

```

```

        "time": 1587906000000,
        "hashrate": "0",
        "reject": 0
    },
    .....
]
},
{
    "workerName": "bhdc1.16A10404B",      //挖矿用户名
    "type": "D_hashrate",                //日均算力类型
    "hashrateDatas": [
        {
            "time": 1587902400000,          // 时间
            "hashrate": "0",                // 算力
            "reject": 0                   //拒绝率
        },
        {
            "time": 1587906000000,
            "hashrate": "0",
            "reject": 0
        },
        .....
    ]
}
]
}

```

`GET /sapi/v1/mining/worker/detail (HMAC SHA256)`

权重(IP): 5

参数:

Name	Type	Mandatory	Description	例子
algo	STRING	YES	算法名称(sha256)	sha256
userName	STRING	YES	挖矿用户名	test
workerName	STRING	YES	矿工用户名，必传	bhdc1.16A10404B
recvWindow	LONG	NO		
timestamp	LONG	YES		

请求矿工列表 (USER_DATA)

响应:

```
{
    "code": 0,
    "msg": "",
    "data": {
        "workerDatas": [
            {

```

```

    "workerId": "1420554439452400131", //矿工id
    "workerName": "2X73", //矿工姓名
    "status": 3, // 状态: 1 有效, 2 无效, 3 失效
    "hashRate": 0, // 实时速率
    "dayHashRate": 0, // 日均算力
    "rejectRate": 0, // 实时拒绝率
    "lastShareTime": 1587712919000 // 最后提交时间
},
{
    "workerId": "7893926126382807951",
    "workerName": "AZDC1.1A10101",
    "status": 2,
    "hashRate": 29711247541680,
    "dayHashRate": 12697781298013.66,
    "rejectRate": 0,
    "lastShareTime": 1587969727000
},
.....
],
"totalNum": 18530, // 总数量
"pageSize": 20 // 每页条数
}
}

```

`GET /sapi/v1/mining/worker/list (HMAC SHA256)`

权重(IP): 5

参数:

Name	Type	Mandatory	Description	例子
algo	STRING	YES	算法名称(sha256)	sha256
userName	STRING	YES	挖矿用户名	test
pageIndex	INTEGER	NO	页码, 为空默认第一页, 从1开始	
sort	INTEGER	NO	排序方向(为空默认为0): 0 正序, 1 倒序	
sortColumn	INTEGER	NO	排序字段(默认为1): 1: 根据矿工名称排序, 2: 根据实时算力排序, 3: 根据日均算力排序, 4: 根据实时拒绝率排序, 5最后提交时间	
workerStatus	INTEGER	NO	矿机状态(默认为0) : 0 全部, 1 有效, 2 无效, 3 失效	
recvWindow	LONG	NO		
timestamp	LONG	YES		

收益列表 (USER_DATA)

响应:

```
{
  "code": 0,
  "msg": "",
  "data": {
    "accountProfits": [
      {
        "time": 1586188800000, // 时间
        "type": 31, // 0:矿池钱包,5:地址挖矿,7:矿池宝,8:已转让,31:收益转让 ,32:算力转让-矿池钱包 33:算力转让-矿池
        "hashTransfer": null, // 已转让算力
        "transferAmount": null, // 已转让收益
        "dayHashRate": 129129903378244, // 算力
        "profitAmount": 8.6083060304, // 奖励数量
        "coinName": "BTC", // 奖励币种
        "status": 2 // 支付状态: 0:待支付, 1:支付中 2: 已支付
      },
      {
        "time": 1607529600000,
        "coinName": "BTC",
        "type": 0,
        "dayHashRate": 9942053925926,
        "profitAmount": 0.85426469,
        "hashTransfer": 200000000000,
        "transferAmount": 0.02180958,
        "status": 2
      },
      {
        "time": 1607443200000,
        "coinName": "BTC",
        "type": 31,
        "dayHashRate": 200000000000,
        "profitAmount": 0.02905916,
        "hashTransfer": null,
        "transferAmount": null,
        "status": 2
      }
    ],
    "totalNum": 3, // 总条数
    "pageSize": 20 // 每页数量
  }
}

```

GET /sapi/v1/mining/payment/list (HMAC SHA256)

权重(IP): 5

参数:

Name	Type	Mandatory	Description	例子
algo	STRING	YES	算法名称(sha256)	sha256
userName	STRING	YES	挖矿用户名	test
coin	STRING	NO	币种名称	
startDate	Long	NO	搜索日期 毫秒时间戳，同时为空查询所有	

Name	Type	Mandatory	Description	例子
endDate	Long	NO	搜索日期 毫秒时间戳，同时为空查询所有	
pageIndex	INTEGER	NO	页码，为空默认第一页，从1开始	
pageSize	INTEGER	NO	分页数量,最小10,最大200	
recvWindow	LONG	NO		
timestamp	LONG	YES		

其他收益列表 (USER_DATA)

响应:

```
{
  "code": 0,
  "msg": "",
  "data": {
    "otherProfits": [
      {
        "time": 1607443200000,           // 时间
        "coinName": "BTC",             // 奖励币种
        "type": 4,                     // 1: 联合挖矿， 2: 活动奖励， 3:返点 4:机枪奖励 6:收益转让 7:矿池宝
        "profitAmount": 0.0011859,     // 奖励数量
        "status": 2                   //支付状态: 0:待支付, 1:支付中 2: 已支付
      }
    ],
    "totalNum": 3,                  // 总条数
    "pageSize": 20                 // 每页数量
  }
}
```

GET /sapi/v1/mining/payment/other (HMAC SHA256)

权重(IP): 5

参数:

Name	Type	Mandatory	Description	例子
algo	STRING	YES	算法名称(sha256)	sha256
userName	STRING	YES	挖矿用户名	test
coin	STRING	NO	币种名称	
startDate	Long	NO	搜索日期 毫秒时间戳，同时为空查询所有	
endDate	Long	NO	搜索日期 毫秒时间戳，同时为空查询所有	
pageIndex	INTEGER	NO	页码，为空默认第一页，从1开始	
pageSize	INTEGER	NO	分页数量,最小10,最大200	

Name	Type	Mandatory	Description	例子
recvWindow	LONG	NO		
timestamp	LONG	YES		

算力转让详情列表 (USER_DATA)

响应:

```
{
  "code": 0,
  "msg": "",
  "data": {
    "configDetails": [
      {
        "configId": 168,      // 该配置的id
        "poolUsername": "123", // 转出子账户
        "toPoolUsername": "user1", // 转入子账户
        "algoName": "Ethash", // 转让算法名称
        "hashRate": 5000000, // 转让算力
        "startDay": 20201210, // 开始时间
        "endDay": 20210405, // 结束时间
        "status": 1          // 状态: 0 进行中, 1: 已取消, 2: 已终止
      },
      {
        "configId": 166,
        "poolUsername": "pop",
        "toPoolUsername": "111111",
        "algoName": "Ethash",
        "hashRate": 3320000,
        "startDay": 20201226,
        "endDay": 20201227,
        "status": 0
      }
    ],
    "totalNum": 21,
    "pageSize": 200
  }
}
```

GET /sapi/v1/mining/hash-transfer/config/details (HMAC SHA256)

权重(IP): 5

参数:

Name	Type	Mandatory	Description	例子
pageIndex	INTEGER	NO	页码, 为空默认第一页, 从1开始	
pageSize	INTEGER	NO	分页数量, 最小10, 最大200	
recvWindow	LONG	NO		

Name	Type	Mandatory	Description	例子
timestamp	LONG	YES		

算力转让列表 (USER_DATA)

响应:

```
{
  "code": 0,
  "msg": "",
  "data": [
    {
      "configDetails": [
        {
          "configId": 168,           // 该配置的id
          "poolUsername": "123",    // 转出子账户
          "toPoolUsername": "user1", // 转入子账户
          "algoName": "Ethash",     // 转让算法名称
          "hashRate": 5000000,       // 转让算力
          "startDay": 20201210,     // 开始时间
          "endDay": 20210405,       // 结束时间
          "status": 1,              // 状态: 0 进行中, 1: 已取消, 2: 已终止
          "type": 0                // 状态: 0 算力转让记录, 1 算力接收记录
        },
        {
          "configId": 166,
          "poolUsername": "pop",
          "toPoolUsername": "111111",
          "algoName": "Ethash",
          "hashRate": 3320000,
          "startDay": 20201226,
          "endDay": 20201227,
          "status": 0,
          "type": 0
        }
      ],
      "totalNum": 21,
      "pageSize": 200
    }
  ]
}
```

GET /sapi/v1/mining/hash-transfer/config/details/list (HMAC SHA256)

权重(IP): 5

参数:

Name	Type	Mandatory	Description	例子
pageIndex	INTEGER	NO	页码, 为空默认第一页, 从1开始	
pageSize	INTEGER	NO	分页数量, 最小10, 最大200	
recvWindow	LONG	NO		

Name	Type	Mandatory	Description	例子
timestamp	LONG	YES		

算力转让详情 (USER_DATA)

响应:

```
{
  "code": 0,
  "msg": "",
  "data": {
    "profitTransferDetails": [
      {
        "poolUsername": "test4001", // 转出子账户
        "toPoolUsername": "pop", // 转入子账户
        "algoName": "sha256", // 转让算法名称
        "hashRate": 200000000000, // 转让算力
        "day": 20201213, // 转让日期
        "amount": 0.2256872, // 转让收益
        "coinName": "BTC" // 收益币种
      },
      {
        "poolUsername": "test4001",
        "toPoolUsername": "pop",
        "algoName": "sha256",
        "hashRate": 200000000000,
        "day": 20201213,
        "amount": 0.2256872,
        "coinName": "BTC"
      }
    ],
    "totalNum": 8,
    "pageSize": 200
  }
}
```

GET /sapi/v1/mining/hash-transfer/profit/details (HMAC SHA256)

权重(IP): 5

参数:

Name	Type	Mandatory	Description	例子
configId	INTEGER	YES	配置的id	168
pageIndex	INTEGER	NO	页码，为空默认第一页，从1开始	
pageSize	INTEGER	NO	分页数量,最小10,最大200	
recvWindow	LONG	NO		
timestamp	LONG	YES		

算力转让请求 (USER_DATA)

响应:

```
{
  "code": 0,
  "msg": "",
  "data": 171 // 该配置的id
}
```

`POST /sapi/v1/mining/hash-transfer/config (HMAC SHA256)`

权重(IP): 5

参数:

Name	Type	Mandatory	Description	例子
userName	STRING	YES	挖矿用户名	test
algo	STRING	YES	算法名称(sha256)	sha256
endDate	Long	YES	转让结束时间(毫秒时间戳)	1617659086000
startDate	Long	YES	转让结束时间(毫秒时间戳)	1607659086000
toPoolUser	STRING	YES	挖矿用户名	S19pro
hashRate	Long	YES	转让算力h/s必传(BTC 大于 500000000000 ETH大于 500000)	100000000
recvWindow	LONG	NO		
timestamp	LONG	YES		

取消算力转让设置 (USER_DATA)

响应:

```
{
  "code": 0,
  "msg": "",
  "data": true
}
```

`POST /sapi/v1/mining/hash-transfer/config/cancel (HMAC SHA256)`

权重(IP): 5

参数:

Name	Type	Mandatory	Description	例子
configId	INTEGER	YES	配置的id	168

Name	Type	Mandatory	Description	例子
userName	STRING	YES	挖矿用户名	test
recvWindow	LONG	NO		
timestamp	LONG	YES		

统计列表 (USER_DATA)

响应:

```
{
  "code": 0,
  "msg": "",
  "data": {
    "fifteenMinHashRate": "457835490067496409.00000000",           // 15分钟算力
    "dayHashRate": "214289268068874127.65000000",           // 日均算力
    "validNum": 0,                                              // 有效数量
    "invalidNum": 17562,                                         // 无效数量
    "profitToday": {
      "BTC": "0.00314332",
      "BSV": "56.17055953",
      "BCH": "106.61586001"
    },
    "profitYesterday": {                                         // 昨日收益
      "BTC": "0.00314332",
      "BSV": "56.17055953",
      "BCH": "106.61586001"
    },
    "userName": "test",                                         // 挖矿账户
    "unit": "h/s",                                            // 算力单位
    "algo": "sha256"                                           // 所属算法
  }
}
```

GET /sapi/v1/mining/statistics/user/status (HMAC SHA256)

权重(IP): 5

参数:

Name	Type	Mandatory	Description	例子
algo	STRING	YES	算法名称(sha256)	sha256
userName	STRING	YES	挖矿用户名	test
recvWindow	LONG	NO		
timestamp	LONG	YES		

账号列表 (USER_DATA)

响应:

```
{
  "code": 0,
  "msg": "",
  "data": [
    {
      "type": "H_hashrate",           //小时算力类型
      "userName": "test",            // 账户名
      "list": [
        {
          "time": 1585267200000,     // 时间
          "hashrate": "0.00000000",   // 算力
          "reject": "0.00000000"     //拒绝率
        },
        {
          "time": 1585353600000,
          "hashrate": "0.00000000",
          "reject": "0.00000000"
        }
        .....
      ]
    },
    {
      "type": "D_hashrate",           //日均算力类型
      "userName": "test",            // 账户名
      "list": [
        {
          "time": 1587906000000,     // 时间
          "hashrate": "0.00000000",   // 算力
          "reject": "0.00000000"     //拒绝率
        },
        {
          "time": 1587909600000,
          "hashrate": "0.00000000",
          "reject": "0.00000000"
        }
        .....
      ]
    }
  ]
}
```

GET /sapi/v1/mining/statistics/user/list (HMAC SHA256)

权重(IP): 5

参数:

Name	Type	Mandatory	Description	例子
algo	STRING	YES	算法名称(sha256)	sha256
userName	STRING	YES	挖矿用户名	test

Name	Type	Mandatory	Description	例子
recvWindow	LONG	NO		
timestamp	LONG	YES		

矿池账户收益列表 (USER_DATA)

响应:

```
{
  "code": 0,
  "msg": "",
  "data": {
    "accountProfits": [
      {
        "time": 1607443200000,           // 时间
        "coinName": "BTC",              // 币种
        "type": 2,                     // 0: 邀请返佣 1: 邀请返现 2: 返点
        "puid": 59985472,              // 挖矿子账户id
        "subName": "vdvaghani",         // 挖矿账户名
        "amount": 0.09186957           // 数量
      }
    ],
    "totalNum": 3,                  // 总条数
    "pageSize": 20                 // 每页数量
  }
}
```

GET /sapi/v1/mining/payment/uid (HMAC SHA256)

权重(IP): 5

参数:

Name	Type	Mandatory	Description	例子
algo	STRING	YES	算法名称(sha256)	sha256
startDate	Long	NO	搜索日期 毫秒时间戳，同时为空查询所有	
endDate	Long	NO	搜索日期 毫秒时间戳，同时为空查询所有	
pageIndex	INTEGER	NO	页码，为空默认第一页，从1开始	
pageSize	INTEGER	NO	分页数量,最小10,最大200	
recvWindow	LONG	NO		
timestamp	LONG	YES		

合约接口

- 列出为了服务于合约产品的接口

合约资金划转 (USER_DATA)

响应:

```
{
  "tranId": 100000001    // 划转 ID
}
```

`POST /sapi/v1/futures/transfer (HMAC SHA256)`

执行现货账户与合约账户之间的划转

权重(IP): 1

参数:

名称	类型	是否必需	描述
asset	STRING	YES	The asset being transferred, e.g., USDT
amount	DECIMAL	YES	The amount to be transferred
type	INT	YES	1: 现货账户向USDT合约账户划转 2: USDT合约账户向现货账户划转 3: 现货账户向币本位合约账户划转 4: 币本位合约账户向现货账户划转
recvWindow	LONG	NO	
timestamp	LONG	YES	

获取合约资金划转历史 (USER_DATA)

响应:

```
{
  "rows": [
    {
      "asset": "USDT",           // 资产
      "tranId": 100000001,       // 划转ID
      "amount": "40.84624400",   // 数量
      "type": "1",               // 划转方向: 1( 现货向USDT本位合约), 2( USDT本位合约向现货), 3( 现货向币本位合
      "timestamp": 1555056425000, // 时间戳
      "status": "CONFIRMED"     // PENDING (等待执行), CONFIRMED (成功划转), FAILED (执行失败);
    }
  ],
  "total": 1
}
```

`GET /sapi/v1/futures/transfer (HMAC SHA256)`

权重(IP): 10

参数:

名称	类型	是否必需	描述
asset	STRING	YES	
startTime	LONG	YES	
endTime	LONG	NO	
current	LONG	NO	当前页面. 起始计数为1. 默认值1
size	LONG	NO	单页数据条目数，默认:10 最大:100
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 仅支持查询最近半年(6个月)数据
- 若 startTime 和 endTime 没传，则默认返回最近7天数据

混合保证金借款历史 (USER_DATA)

响应:

```
{
  "rows": [
    {
      "confirmedTime": 1582540328433,
      "coin": "USDT",
      "collateralRate": "0.89991001", // collateralLevel
      "leftTotal": "4.5",
      "leftPrincipal": "4.5",
      "deadline": 4736102399000,
      "collateralCoin": "BUSD",
      "collateralAmount": "5.0",
      "orderStatus": "PENDING",
      "borrowId": "438648398970089472"
    }
  ],
  "total": 1
}
```

`GET /sapi/v1/futures/loan/borrow/history (HMAC SHA256)`

参数:

名称	类型	是否必需	描述

名称	类型	是否必需	描述
coin	STRING	NO	
startTime	LONG	NO	
endTime	LONG	NO	
limit	LONG	NO	default 500, max 1000
recvWindow	LONG	NO	
timestamp	LONG	YES	

权重(IP): 10

混合保证金还款历史 (USER_DATA)

相应:

```
{
  "rows": [
    {
      "coin": "USDT",
      "amount": "1.68",
      "collateralCoin": "BUSD",
      "repayType": "NORMAL", // "COLLATERAL" 则为抵押物还款
      "releasedCollateral": "1.80288889",
      "price": "1.001", // 借贷/抵押物还款兑换价格比率
      "repayCollateral": "10010", // 抵押物还款所用抵押物量
      "confirmedTime": 1582781327575,
      "updateTime": 1582794387516, // 时间
      "status": "PENDING",
      "repayId": "439659223998894080"
    }
  ],
  "total": 1
}
```

GET /sapi/v1/futures/loan/repay/history HMAC SHA256)

参数:

名称	类型	是否必需	描述
coin	STRING	NO	
startTime	LONG	NO	
endTime	LONG	NO	
limit	LONG	NO	default 500, max 1000
recvWindow	LONG	NO	

名称	类型	是否必需	描述
timestamp	LONG	YES	

权重(IP): 10

混合保证金钱包V2 (USER_DATA)

响应:

```
{
    "totalCrossCollateral": "5.8238577133",
    "totalBorrowed": "5.07000000",
    "totalInterest": "0.0", // 混合保证金总利息
    "interestFreeLimit": "100000", // 混合保证金总免息额度
    "asset": "USD", // USD 计价
    "crossCollaterals": [
        {
            "loanCoin": "USDT",
            "collateralCoin": "BUSD",
            "locked": "5.82211108",
            "loanAmount": "5.07",
            "currentCollateralRate": "0.87168984", // collateralLevel
            "interestFreeLimitUsed": "5.07", // 占用混合保证金免息额度
            "principalForInterest": "0.0", // 混合保证金利息计算所用本金
            "interest": "0.0" // 混合保证金利息
        },
        {
            "loanCoin": "BUSD",
            "collateralCoin": "BTC",
            "locked": "0",
            "loanAmount": "0",
            "currentCollateralRate": "0", // collateralLevel
            "interestFreeLimitUsed": "0", // 占用混合保证金免息额度
            "principalForInterest": "0.0", // 混合保证金利息计算所用本金
            "interest": "0.0" // 混合保证金利息
        }
    ]
}
```

GET /sapi/v2/futures/loan/wallet (HMAC SHA256)

参数:

名称	类型	是否必需	描述
recvWindow	LONG	NO	
timestamp	LONG	YES	

权重(IP): 1

混合保证金调整质押率历史 (USER_DATA)

响应:

```
{
  "rows": [
    {
      "amount": ".17398184",
      "collateralCoin": "BUSD",
      "coin": "USDT",
      "preCollateralRate": "0.87054861",
      "afterCollateralRate": "0.89736451",
      "direction": "REDUCED",
      "status": "COMPLETED",
      "adjustTime": 1583978243588
    }
  ],
  "total": 1
}
```

GET /sapi/v1/futures/loan/adjustCollateral/history (HMAC SHA256)

参数:

Name	Type	Mandatory	Description
loanCoin	STRING	NO	
collateralCoin	STRING	NO	
startTime	LONG	NO	
endTime	LONG	NO	
limit	LONG	NO	default 500, max 1000
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 如果不传"loanCoin"或"collateralCoin"返回所有借币和抵押物信息

权重(IP): 10

混合保证金强平历史 (USER_DATA)

响应:

```
{
  "rows": [
    {
      "collateralAmountForLiquidation": "10.12345678",
      "collateralCoin": "BUSD",
      "liquidationTime": 1583978243588
    }
  ],
  "total": 1
}
```

```

        "forceLiquidationStartTime": 1583978243588,
        "coin": "USDT",
        "restCollateralAmountAfterLiquidation": "15.12345678",
        "restLoanAmount": "11.12345678",
        "status": "PENDING"
    },
],
"total": 1
}

```

GET /sapi/v1/futures/loan/liquidationHistory (HMAC SHA256)

参数:

名称	类型	是否必需	描述
loanCoin	STRING	NO	
collateralCoin	STRING	NO	
startTime	LONG	NO	
endTime	LONG	NO	
limit	LONG	NO	default 500, max 1000
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 如果不传"loanCoin"或"collateralCoin"返回所有借币和抵押物信息

权重(IP): 10

混合保证金利息收取历史 (USER_DATA)

响应:

```

{
  "rows": [
    {
      "collateralCoin": "BUSD",
      "interestCoin": "USDT",
      "interest": "2.354",
      "interestFreeLimitUsed": "0", // 占用混合保证金免息额度
      "principalForInterest": "10000",
      "interestRate": "0.002",
      "time": 1582794387516
    }
  ],
  "total": 1
}

```

GET /sapi/v1/futures/loan/interestHistory (HMAC SHA256)

参数:

名称	类型	是否必需	描述
collateralCoin	STRING	NO	
startTime	LONG	NO	
endTime	LONG	NO	
current	LONG	NO	当前查询页。开始值 1。默认:1
limit	LONG	NO	默认 500 , 最大 1000
recvWindow	LONG	NO	
timestamp	LONG	YES	

权重(IP): 1

合约策略交易接口

币安合约算法交易API服务，旨在为用户提供一整套的算法交易解决方案，包括：自动执行订单，提高执行透明度和提供智能接口直达市场。

FAQ: 成交量份额参与算法(VP) 介绍

FAQ: 时间加权平均价格策略(Twap) 介绍

成交量份额参与算法(VP)下单 (TRADE)

响应:

```
{
  "clientAlgoId": "00358ce6a268403398bd34eaa36dffe7", // 用户自定义策略订单ID
  "success": true,
  "code": 0,
  "msg": "OK"
}
```

POST /sapi/v1/algo/futures/newOrderVp (HMAC SHA256)

发送VP下单 仅支持U本位合约

权重(UID): 3000**注意:**

- 您的 API Key 需要开通 **允许合约交易** 权限
- 请使用Base URL: <https://api.binance.com>

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	交易对 eg. BTCUSDT
side	ENUM	YES	买卖方向 (BUY or SELL)
positionSide	ENUM	NO	持仓方向，单向持仓模式下非必填，默认且仅可填BOTH;在双向持仓模式下必填,且仅可选择LONG 或 SHORT
quantity	DECIMAL	YES	下单数量，以合约币种 (base asset) 个数下单; 名义价值 (quantity) * 标记价格(base asset) 需要大于 10,000 USDT , 且不超过 1,000,000 USDT
urgency	ENUM	YES	代表当前执行的相对速率; ENUM: LOW (慢), MEDIUM (中等), HIGH (快)
clientAlgoId	STRING	NO	必须传入32位，如果未发送，则自动生成
reduceOnly	BOOLEAN	NO	true, false; 非双开模式下默认false；双开模式下不接受此参数；开仓不接受此参数
limitPrice	DECIMAL	NO	限价单价格; 若未发送，则以市场价下单
recvWindow	LONG	NO	
timestamp	LONG	YES	

其他信息:

- 最大所有策略订单挂单数量：10。
- 杠杆倍数和持仓模式与您的合约账户设置相同，您可以通过合约交易页面设置或者通过fapi设置。
- 收到 "success": true 不代表您的订单一定会被执行。请通过查询订单接口 (GET sapi/v1/algo/futures/openOrders) 或者 (GET sapi/v1/algo/futures/historicalOrders) 以获取订单状态。例如: 如果您的合约账户余额不足，或者开仓使用了 reduce only 参数，或者您下单选择的持仓模式与您设置的不符，这些情况您都会收到响应 "success": true，但订单状态会显示为 expired，代表订单过期。

时间加权平均价格策略(Twap)下单 (TRADE)

响应:

```
{
  "clientAlgoId": "65ce1630101a480b85915d7e11fd5078", // 用户自定义策略订单ID
  "success": true,
  "code": 0,
  "msg": "OK"
}
```

POST /sapi/v1/algo/futures/newOrderTwap (HMAC SHA256)

发送Twap下单 仅支持U本位合约

权重(UID): 3000

注意:

- 您的 API Key 需要开通 允许合约交易 权限
- 请使用Base URL: <https://api.binance.com>

参数:

名称	类型	是否必需	描述
symbol	STRING	YES	交易对 eg. BTCUSDT
side	ENUM	YES	买卖方向 (BUY or SELL)
positionSide	ENUM	NO	持仓方向，单向持仓模式下非必填，默认且仅可填BOTH;在双向持仓模式下必填且仅可选择LONG 或 SHORT
quantity	DECIMAL	YES	下单数量，以合约币种 (base asset) 个数下单; 名义价值 (<code>quantity</code> * 标记价格(base asset)) 需要大于 10,000 USDT，且不超过 1,000,000 USDT
duration	LONG	YES	请以秒为单位发送[300,86400]；少于 5 分钟 => 默认为 5 分钟；大于 24h => 默认为 24h
clientAlgoId	STRING	NO	必须传入32位，如果未发送，则自动生成
reduceOnly	BOOLEAN	NO	true, false; 非双开模式下默认false；双开模式下不接受此参数；开仓不接受此参数
limitPrice	DECIMAL	NO	限价单价格, 若未发送，则以市场价下单
recvWindow	LONG	NO	
timestamp	LONG	YES	

其他信息:

- 最大所有策略订单挂单数量：10。
- 杠杆倍数和持仓模式与您的合约账户设置相同，您可以通过合约交易页面设置或者通过fapi设置。
- 收到 "success": true 不代表您的订单一定会被执行。请通过查询订单接口 (GET sapi/v1/algo/futures/openOrders 或者 GET sapi/v1/algo/futures/historicalOrders) 以获取订单状态。例如: 如果您的合约账户余额不足，或者开仓使用了 reduce only 参数，或者您下单选择的持仓模式与您设置的不符，这些情况您都会收到响应 "success": true，但订单状态会显示为 expired，代表订单过期。
- `quantity` * 60 / `duration` 必须大于minQty。
- `duration` 不能小于5分钟，且不能大于24小时。
- 对于U本位交割合约, TWAP 的结束时间必须早于交割时间1小时。

取消策略订单 (TRADE)

响应:

```
{
  "algoId": 14511, //策略订单ID
  "success": true,
  "code": 0,
  "msg": "OK"
}
```

`DELETE /sapi/v1/algo/futures/order (HMAC SHA256)`

撤销订单

权重(IP): 1

注意:

- 您的 API Key 需要开通 允许合约交易 权限

- 请使用Base URL: <https://api.binance.com>

参数:

名称	类型	是否必需	描述
algoId	LONG	YES	eg. 14511
recvWindow	LONG	NO	
timestamp	LONG	YES	

查询当前策略订单挂单 (USER_DATA)

响应:

```
{
  "total": 1,
  "orders": [
    {
      "algoId": 14517,           //策略订单ID
      "symbol": "ETHUSDT",      //交易对
      "side": "SELL",          //买卖方向
      "positionSide": "SHORT", //持仓模式
      "totalQty": "5.000",      //总共下单数量
      "executedQty": "0.000",   //执行数量
      "executedAmt": "0.00000000", //执行价值
      "avgPrice": "0.00",       //平均价格
      "clientAlgoId": "d7096549481642f8a0bb69e9e2e31f2e", //用户自定义策略订单ID
      "bookTime": 1649756817004, //用户下单时间
      "endTime": 0,             //结束时间
      "algoStatus": "WORKING", //策略订单状态
      "algoType": "VP",         //策略订单类型
      "urgency": "LOW"          //执行速率
    }
  ]
}
```

GET /sapi/v1/algo/futures/openOrders (HMAC SHA256)

权重(IP): 1**注意:**

- 您的 API Key 需要开通 允许合约交易 权限
- 请使用Base URL: <https://api.binance.com>

参数:

名称	类型	是否必需	描述
recvWindow	LONG	NO	
timestamp	LONG	YES	

查询历史策略订单 (USER_DATA)

响应:

```
{
  "total": 1,
  "orders": [
    {
      "algoId": 14518,           //策略订单ID
      "symbol": "BNBUSDT",      //交易对
      "side": "BUY",            //买卖方向
      "positionSide": "BOTH",   //持仓模式
      "totalQty": "100.00",     //总共下单数量
      "executedQty": "0.00",    //执行数量
      "executedAmt": "0.00000000", //执行价值
      "avgPrice": "0.000",       //平均价格
      "clientAlgoId": "acacab56b3c44bef9f6a8f8ebd2a8408", //用户自定义策略订单ID
      "bookTime": 1649757019503, //用户下单时间
      "endTime": 1649757088101, //结束时间
      "algoStatus": "CANCELLED", //策略订单状态
      "algoType": "VP",          //策略订单类型
      "urgency": "LOW"           //执行速率
    }
  ]
}
```

GET /sapi/v1/algo/futures/historicalOrders (HMAC SHA256)

权重(IP): 1

注意:

- 您的 API Key 需要开通 **允许合约交易** 权限
- 请使用Base URL: <https://api.binance.com>

参数:

名称	类型	是否必需	描述
symbol	STRING	NO	交易对 eg. BTCUSDT
side	ENUM	NO	BUY 或者 SELL
startTime	LONG	NO	毫秒级时间戳 eg.1641522717552
endTime	LONG	NO	毫秒级时间戳 eg.1641522526562
page	INT	NO	默认 1
pageSize	INT	NO	最小 1, 最大 100; 默认 100
recvWindow	LONG	NO	
timestamp	LONG	YES	

查询执行子订单 (USER_DATA)

响应:

```
{
  "total": 1,
  "executedQty": "1.000",
  "executedAmt": "3229.44000000",
  "subOrders": [
    {
      "algoId": 13723,          //策略订单ID
      "orderId": 8389765519993908929, //子订单ID
      "orderStatus": "FILLED",      //子订单状态
      "executedQty": "1.000",      //执行数量
      "executedAmt": "3229.44000000", //执行价值
      "feeAmt": "-1.61471999",     //手续费
      "feeAsset": "USDT",         //手续费币种
      "bookTime": 1649319001964,   //下单时间
      "avgPrice": "3229.44",       //平均价格
      "side": "SELL",            //买卖方向
      "symbol": "ETHUSDT",        //交易对
      "subId": 1,                 //子订单执行顺序ID
      "timeInForce": "IMMEDIATE_OR_CANCEL", //有效方式
      "origQty": "1.000"          //原始委托数量
    }
  ]
}
```

GET /sapi/v1/algo/futures/subOrders (HMAC SHA256)

获取指定 algoId 的相应子订单

权重(IP): 1

注意:

- 您的 API Key 需要开通 允许合约交易 权限
- 请使用Base URL: <https://api.binance.com>

参数:

名称	类型	是否必需	描述
algoId	LONG	YES	
page	INT	NO	默认1
pageSize	INT	NO	最小 1 , 最大 100; 默认 100
recvWindow	LONG	NO	
timestamp	LONG	YES	

统一账户接口

为了给币安合约用户提供更加优质的服务及提高用户的资金利用率，币安将推出统一账户计划。该计划将以合约钱包、现货杠杆钱包的总资产作为保证金来计算。关于统一账户：币安统一账户计划是一项跨资产保证金计划，支持超过 200 种有效的加密资产。U本位合约、币本位合约以及杠杆钱包中支持的加密资产和头寸将作为有效的联合抵押品，以确定统一账户的权益、保证金余额和维持保证金要求。

FAQ: 币安合约统一账户总览

仅对特定用户开放此功能，详情：加入统一账户计划

查询统一账户信息 (USER_DATA)

响应:

```
{
    "uniMMR": "5167.92171923",           // 统一账户模式维持保证金率
    "accountEquity": "122607.35137903",   // 统一账户总权益，单位为USD
    "accountMaintMargin": "23.72469206", // 统一账户维持保证金，即账户开仓及借贷总共需要的维持保证金，单位为
    "accountStatus": "NORMAL"          // 统一账户当前账户状态："NORMAL"正常状态，"MARGIN_CALL"补充保证金，"SUPPLY_
}
```

GET /sapi/v1/portfolio/account (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
recvWindow	LONG	NO	
timestamp	LONG	YES	

统一账户资产质押率 (MARKET_DATA)

响应 :

```
[
    {
        "asset": "USDC",
        "collateralRate": "1.0000" //质押率
    },
    {
        "asset": "BUSD",
        "collateralRate": "1.0000"
    },
]
```

GET /sapi/v1/portfolio/collateralRate

<https://binance-docs.github.io/apidocs/spot/cn/#185368440e>

237/295

统一账户资产质押率

权重(IP): 50

参数 : None

查询统一账户穿仓借贷金额 (USER_DATA)

响应 :

```
{
    "asset": "BUSD",
    "amount": "579.45", // 统一账户用户强平穿仓负债，单位为BUSD
}
```

`GET /sapi/v1/portfolio/pmLoan`

查询统一账户穿仓借贷金额

权重(UID): 500

参数 :

名称	类型	是否必需	描述
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 如果不存在统一账户穿仓负债，amount显示为0

偿还统一账户穿仓负债

响应 :

```
{
    "tranId": 58203331886213504
}
```

`POST /sapi/v1/portfolio/repay`

偿还统一账户穿仓负债

权重(UID): 3000

参数 :

名称	类型	是否必需	描述
recvWindow	LONG	NO	
timestamp	LONG	YES	

杠杆代币接口

杠杆代币信息 (MARKET_DATA)

响应:

```
[  
 {  
   "tokenName": "BTCDOWN",  
   "description": "3X Short Bitcoin Token",  
   "underlying": "BTC",  
   "tokenIssued": "717953.95",  
   "basket": "-821.474 BTCUSDT Futures",  
   "currentBaskets": [  
     {  
       "symbol": "BTCUSDT",  
       "amount": "-1183.984",  
       "notionalValue": "-22871089.96704"  
     }  
   ],  
   "nav": "4.79",  
   "realLeverage": "-2.316",  
   "fundingRate": "0.001020",  
   "dailyManagementFee": "0.0001",  
   "purchaseFeePct": "0.0010", // 申购费率  
   "dailyPurchaseLimit": "100000.00", // 每日申购数量上限  
   "redeemFeePct": "0.0010", // 赎回费率  
   "dailyRedeemLimit": "1000000.00", // 每日赎回数量上限  
   "timestamp": 1583127900000  
 },  
 {  
   "tokenName": "LINKUP",  
   "description": "3X LONG ChainLink Token",  
   "underlying": "LINK",  
   "tokenIssued": "163846.99",  
   "basket": "417288.870 LINKUSDT Futures",  
   "currentBaskets": [  
     {  
       "symbol": "LINKUSDT",  
       "amount": "1640883.83",  
       "notionalValue": "22596611.22293"  
     }  
   ],  
   "nav": "9.60",  
   "realLeverage": "2.597",  
   "fundingRate": "-0.000917",  
   "dailyManagementFee": "0.0001",  
   "purchaseFeePct": "0.0010",  
   "dailyPurchaseLimit": "100000.00",  
   "redeemFeePct": "0.0010",  
   "dailyRedeemLimit": "1000000.00",  
   "timestamp": 1583127900000  
 }
```

},
]

GET /sapi/v1/blvt/tokenInfo

权重(IP): 1**参数:**

名称	类型	是否必需	描述
tokenName	STRING	NO	BTCDOWN, BTCUP

杠杆代币历史净值K线

杠杆代币净值系统基于合约架构，故该接口采用fapi

请前往[这里](#)查看相关接口，并按照fapi使用规范操作。

申购代币 (USER_DATA)

响应:

```
{
  "id": 123,
  "status": "S", // S, P, F 分别表示 "success", "pending", "failure"
  "tokenName": "LINKUP",
  "amount": "0.95590905", // 申购代币数量
  "cost": "9.99999995", // 申购金额
  "timestamp": 1600249972899
}
```

POST /sapi/v1/blvt/subscribe (HMAC SHA256)

权重(IP): 1**参数:**

名称	类型	是否必需	描述
tokenName	STRING	YES	BTCDOWN, BTCUP
cost	DECIMAL	YES	
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 您需要为API Key开通[允许现货和杠杆交易](#)权限才能发送此请求

查询申购记录 (USER_DATA)

响应:

```
[
  [
    {
      "id": 1,
      "tokenName": "LINKUP",
      "amount": "0.54216292",
      "nav": "18.42621386", // usdt计价的申购净值
      "fee": "0.00999000", // usdt计价的申购费用
      "totalCharge": "9.99999991", //usdt计价的申购总金额
      "timestamp":1599127217916
    }
  ]
]
```

GET /sapi/v1/blvt/subscribe/record (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
tokenName	STRING	NO	BTCDOWN, BTCUP
id	LONG	NO	
startTime	LONG	NO	
endTime	LONG	NO	
limit	INT	NO	默认 1000, 最大 1000
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 只可查询最近90天记录

赎回代币 (USER_DATA)

响应:

```
{
  "id": 123,
  "status": "S", // S, P, F 分别表示 "success", "pending", "failure"
  "tokenName": "LINKUP",
  "redeemAmount": "0.95590905", // 赎回代币数量
  "amount": "10.05022099", // usdt计价的赎回金额
  "timestamp":1600250279614
}
```

`POST /sapi/v1/blvt/redeem (HMAC SHA256)`

权重(IP): 1

参数:

名称	类型	是否必需	描述
tokenName	STRING	YES	BTCDOWN, BTCUP
amount	DECIMAL	YES	
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 您需要为API Key开通允许现货和杠杆交易权限才能发送此请求

查询赎回记录 (USER_DATA)

响应:

```
[
  {
    "id": 1,
    "tokenName": "LINKUP",
    "amount": "0.54216292", // 赎回数量
    "nav": "18.36345064", // usdt计价的赎回净值
    "fee": "0.00995598", // usdt计价的赎回费用
    "netProceed": "9.94602604", // usdt计价的净赎回金额
    "timestamp": 1599128003050
  }
]
```

`GET /sapi/v1/blvt/redeem/record (HMAC SHA256)`

权重(IP): 1

参数:

名称	类型	是否必需	描述
tokenName	STRING	NO	BTCDOWN, BTCUP
id	LONG	NO	
startTime	LONG	NO	
endTime	LONG	NO	
limit	INT	NO	默认 1000, 最大 1000
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 只可查询最近90天记录

查询用户每日申购赎回限额 (USER_DATA)

响应:

```
[
  {
    "tokenName": "LINKUP",
    "userDailyTotalPurchaseLimit": "1000", // 用户个人每日申购数量上限 (USDT)
    "userDailyTotalRedeemLimit": "1000"   // 用户个人每日赎回数量上限 (USDT)
  },
  {
    "tokenName": "LINKDOWN",
    "userDailyTotalPurchaseLimit": "1000", // 用户个人每日申购数量上限 (USDT)
    "userDailyTotalRedeemLimit": "50000"  // 用户个人每日赎回数量上限 (USDT)
  }
]
```

GET /sapi/v1/blvt/userLimit (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
tokenName	STRING	NO	BTCDOWN, BTCUP
recvWindow	LONG	NO	
timestamp	LONG	YES	

WebSocket 杠杆代币信息更新

Payload:

```
{
  "e": "nav",           // 事件类型
  "E": 1600245286355, // 事件时间
  "s": "TRXDOWN",     // 杠杆代币
  "m": 74164.75496502663, // 已发行代币
  "b": [               // 篮子
    {
      "s": "TRXUSDT", // 合约交易对
      "n": -87988261 // 持仓数量
    }
  ],
  "n": 14.78454447, // 净值
  "l": 2.1786579638117898, // 真实杠杆
  "t": 3,             // 目标杠杆
}
```

```
"f": -0.0048925 // 资金费率
}
```

Stream Name: <tokenName>@tokenNav

- 注意: 您需要使用 `wss://nbstream.binance.com/lvt-p` 来监听该数据流
- 注意: tokenName 请使用**大写字母**, 例如"TRXDOWN@tokenNav"

Update Speed: 3s

WebSocket 杠杆代币净值K线更新

Payload:

```
{
  "e": "kline", // 事件类型
  "E": 1600243159447, // 事件时间
  "s": "TRXDOWN", // 杠杆代币
  "k": {
    "t": 1600243140000, // 这根K线的起始时间
    "T": 1600243199999, // 这根K线的结束时间
    "s": "TRXDOWN", // 杠杆代币
    "i": "1m", // K线间隔
    "f": 1600243140484, // 这根K线期间第一笔净值更新时间
    "L": 1600243159424, // 这根K线期间末一笔净值更新时间
    "o": "14.56800297", // 这根K线期间第一笔净值
    "c": "14.59766270", // 这根K线期间末一笔净值
    "h": "14.63325437", // 这根K线期间最高净值
    "l": "14.56207102", // 这根K线期间最低净值
    "v": "2.22524220", // 真实杠杆倍数
    "n": 33, // 这根K线期间更新的净值次数
    "x": false, // 忽略此参数
    "q": "0", // 忽略此参数
    "V": "73.42663923", // 忽略此参数
    "Q": "0", // 忽略此参数
    "B": "0" // 忽略此参数
  }
}
```

Stream Name: <tokenName>@nav_kline_<interval>

- 注意: 您需要使用 `wss://nbstream.binance.com/lvt-p` 来监听该数据流
- 注意: tokenName 请使用**大写字母**, 例如"TRXDOWN@nav_kline_1d"

Update Speed: 300ms**K线图间隔参数:**

m -> 分钟; h -> 小时; d -> 天; w -> 周; M -> 月

- 1m
- 3m
- 5m
- 15m
- 30m
- 1h
- 2h

- 4h
- 6h
- 8h
- 12h
- 1d
- 3d
- 1w
- 1M

币安挖矿接口

- 这些接口用于币安挖矿产品。更多细节, 请参考币安挖矿页面。

获取所有流动资金池 (MARKET_DATA)

响应:

```
[
  {
    "poolId": 2,
    "poolName": "BUSD/USDT",
    "assets": [
      "BUSD",
      "USDT"
    ]
  },
  {
    "poolId": 3,
    "poolName": "BUSD/DAI",
    "assets": [
      "BUSD",
      "DAI"
    ]
  },
  {
    "poolId": 4,
    "poolName": "USDT/DAI",
    "assets": [
      "USDT",
      "DAI"
    ]
  }
]
```

`GET /sapi/v1/bswap/pools`

获取币安挖矿产品中所有资金池

权重(IP): 1

参数: 无

获取流动资金池具体信息 (USER_DATA)

响应:

```
[
  {
    "poolId": 2,
    "poolName": "BUSD/USDT",
    "updateTime": 1565769342148,
    "liquidity": {
      "BUSD": 100000315.79,
      "USDT": 99999245.54
    },
    "share": {
      "shareAmount": 12415,
      "sharePercentage": 0.00006207,
      "asset": {
        "BUSD": 6207.02,
        "USDT": 6206.95
      }
    }
  }
]
```

GET /sapi/v1-bswap/liquidity (HMAC SHA256)

获取某个流动资金池具体信息

权重(IP):

1 单一池

10 poolId 参数缺失

频次限制 : 每个账户每个池子最多每秒三次

参数:

名称	类型	是否必需	描述
poolId	LONG	NO	
recvWindow	LONG	NO	
timestamp	LONG	YES	

添加流动性 (TRADE)

响应:

```
{
  "operationId": 12341
}
```

POST /sapi/v1-bswap/liquidityAdd (HMAC SHA256)

向某个资金池添加流动性

权重(UID): 1000 (额外限制 : 每3秒一次)

参数:

名称	类型	是否必需	描述
poolId	LONG	YES	
type	STRING	NO	"SINGLE" 为单币添加资产; "COMBINATION" 为双币添加资产。 默认 "SINGLE"
asset	STRING	YES	
quantity	DECIMAL	YES	
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 您需要为API Key开通**允许现货和杠杆交易**权限才能发送此请求

移除流动性 (TRADE)

响应:

```
{
  "operationId": 12341
}
```

POST /sapi/v1-bswap/liquidityRemove (HMAC SHA256)

从某个资金池移除流动性 , **type** 包含 **SINGLE** 和 **COMBINATION** , 如果是单币种移除则 **asset** 参数为必需

权重(UID): 1000 (额外限制 : 每3秒一次)

参数:

名称	类型	是否必需	描述
poolId	LONG	YES	
type	STRING	YES	SINGLE 为以单币种移出, COMBINATION 为以池中所有币种按比例移出
asset	LIST	NO	如果是单币种移除则 asset 参数为必需
shareAmount	DECIMAL	YES	
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 您需要为API Key开通**允许现货和杠杆交易**权限才能发送此请求

获取流动性操作记录 (USER_DATA)

响应:

```
[
  {
    "operationId": 12341,
    "poolId": 2,
    "poolName": "BUSD/USDT",
    "operation": "ADD", // "ADD" or "REMOVE"
    "status": 1, // 0: pending, 1: success, 2: failed
    "updateTime": 1565769342148,
    "shareAmount": "10.1"
  }
]
```

`GET /sapi/v1-bswap/liquidityOps (HMAC SHA256)`

获取流动性操作 (添加/移除) 记录

权重(UID): 3000

参数:

名称	类型	是否必需	描述
operationId	LONG	NO	
poolId	LONG	NO	
operation	ENUM	NO	<code>ADD</code> 或 <code>REMOVE</code>
startTime	LONG	NO	
endTime	LONG	NO	
limit	LONG	NO	默认 3, 最大 100
recvWindow	LONG	NO	
timestamp	LONG	YES	

获取报价 (USER_DATA)

响应:

```
{
  "quoteAsset": "USDT",
  "baseAsset": "BUSD",
  "quoteQty": 300000,
  "baseQty": 299975,
  "price": 1.00008334,
  "slippage": 0.00007245,
```

```

    "fee": 120
}

```

GET /sapi/v1-bswap/quote (HMAC SHA256)

通过传入计价币（卖出币）与交易币（买入币）获取报价，即给定量的兑换汇率。`quoteQty` 为卖出币数量。注意该报价为参考，实际会随流动性变动而变化，推荐在获取报价之后立刻进行兑换交易以避免滑点变化过大。

权重(UID): 150

频次限制： 每个账户每个池子最多每秒三次

参数:

名称	类型	是否必需	描述
quoteAsset	STRING	YES	
baseAsset	STRING	YES	
quoteQty	DECIMAL	YES	
recvWindow	LONG	NO	
timestamp	LONG	YES	

交易 (TRADE)

响应:

```

{
    "swapId": 2314
}

```

POST /sapi/v1-bswap/swap (HMAC SHA256)

交易，即兑换 `quoteAsset` 为 `baseAsset`。

权重(UID): 1000 (额外限制 : 每2秒一次)

参数:

名称	类型	是否必需	描述
quoteAsset	STRING	YES	
baseAsset	STRING	YES	
quoteQty	DECIMAL	YES	
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 您需要为API Key开通 `允许现货和杠杆交易` 权限才能发送此请求

获取交易记录 (USER_DATA)

响应:

```
[  
  {  
    "swapId": 2314,  
    "swapTime": 1565770342148,  
    "status": 0, // 0: pending, 1: success, 2: failed  
    "quoteAsset": "USDT",  
    "baseAsset": "BUSD",  
    "quoteQty": 300000,  
    "baseQty": 299975,  
    "price": 1.00008334,  
    "fee": 120  
  }  
]
```

GET /sapi/v1/bswap/swap (HMAC SHA256)

获取交易历史记录。

权重(UID): 3000

参数:

名称	类型	是否必需	描述
swapId	LONG	NO	
startTime	LONG	NO	
endTime	LONG	NO	
status	INT	NO	0: 交易中, 1: 交易成功, 2: 交易失败
quoteAsset	STRING	NO	
baseAsset	STRING	NO	
limit	LONG	NO	默认 3, 最大 100
recvWindow	LONG	NO	
timestamp	LONG	YES	

获取币对池的配置信息 (USER_DATA)

响应:

```
[  
  {  
    "poolId": 2,  
    "poolName": "BUSD/USDT",  
  }
```

```

    "updateTime": 1565769342148,
    "liquidity": {
        "constantA": 2000, //如果为创新池则展示“NA”
        "minRedeemShare": 0.1,
        "slippageTolerance": 0.2 //仅当滑点在区间内才执行交易
    },
    "assetConfigure": {
        "BUSD": {
            "minAdd": 10,
            "maxAdd": 20,
            "minSwap": 10,
            "maxSwap": 30
        },
        "USDT": {
            "minAdd": 10,
            "maxAdd": 20,
            "minSwap": 10,
            "maxSwap": 30
        }
    }
}
]

```

GET /sapi/v1-bswap/poolConfigure (HMAC SHA256)

权重(IP): 150

参数:

名称	类型	是否必需	描述
poolId	LONG	NO	
recvWindow	LONG	NO	
timestamp	LONG	YES	

添加流动性的试算 (USER_DATA)

响应:

```
{
    "quoteAsset": "USDT",
    "baseAsset": "BUSD", //类型为"COMBINATION"时展示
    "quoteAmt": 300000,
    "baseAmt": 299975, // 类型为"COMBINATION"时展示
    "price": 1.00008334,
    "share": 1.23,
    "slippage": 0.00007245, //类型为"SINGLE"时展示
    "fee": 120, // 类型为"SINGLE"时展示
}
```

GET /sapi/v1-bswap/addLiquidityPreview (HMAC SHA256)

计算查询单币添加或双币添加流动性需要的资产数量及获得的份额

权重(IP): 150

参数:

名称	类型	是否必需	描述
poolId	LONG	YES	
type	STRING	YES	类型为"SINGLE"意思为单币添加;类型为"COMBINATION"意思为双币添加
quoteAsset	STRING	YES	
quoteQty	DECIMAL	YES	
recvWindow	LONG	NO	
timestamp	LONG	YES	

移除流动性的试算 (USER_DATA)

响应:

```
{
    "quoteAsset": "USDT",
    "baseAsset": "BUSD", // 类型为"COMBINATION"时展示
    "quoteAmt": 300000,
    "baseAmt": 299975, //类型为"COMBINATION"时展示
    "price": 1.00008334,
    "slippage": 0.00007245, //类型为"SINGLE"时展示
    "fee": 120 //类型为"SINGLE"时展示
}
```

GET /sapi/v1/bswap/removeLiquidityPreview (HMAC SHA256)

计算查询单币赎回或双币赎回预计获得的资产数量

权重(IP): 150

参数:

名称	类型	是否必需	描述
poolId	LONG	YES	
type	STRING	YES	类型为"SINGLE"意思为移除获得单币；类型为"COMBINATION"意思为移除获得双币
quoteAsset	STRING	YES	
shareAmount	DECIMAL	YES	
recvWindow	LONG	NO	
timestamp	LONG	YES	

查询未领取的奖励数量 (USER_DATA)

响应:

```
{
    "totalUnclaimedRewards": {
        "BUSD": 100000315.79,
        "BNB": 0.0000001,
        "USDT": 0.0000002
    },
    "details": {
        "BNB/USDT": {
            "BUSD": 100000315.79,
            "USDT": 0.0000002
        },
        "BNB/BTC": {
            "BNB": 0.0000001
        }
    }
}
```

GET /sapi/v1-bswap/unclaimedRewards (HMAC SHA256)

查询未领取的奖励数量

权重(UID): 1000

参数:

名称	类型	是否必需	描述
type	INT	NO	0:交易挖矿奖励，1:流动性挖矿奖励，默认为0
recvWindow	LONG	NO	
timestamp	LONG	YES	

领取奖励 (TRADE)

响应:

```
{
    "success":true
}
```

POST /sapi/v1-bswap/claimRewards (HMAC SHA256)

领取交易挖矿奖励或者流动性挖矿奖励

权重(UID): 1000

参数:

名称	类型	是否必需	描述
type	INT	NO	0:交易挖矿奖励 , 1:流动性挖矿奖励 , 默认为0
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 您需要为API Key开通允许现货和杠杆交易权限才能发送此请求

获取已领取奖励记录 (USER_DATA)

响应:

```
[
  {
    "poolId":52,
    "poolName": "BNB/USDT",
    "assetRewards": "BNB",
    "claimTime":1565769342148,
    "claimAmount":0.00000023,
    "status":1 // 0: pending, 1: success
  }
]
```

GET /sapi/v1-bswap/claimedHistory (HMAC SHA256)

获取已领取奖励的历史记录

权重(UID): 1000

参数:

名称	类型	是否必需	描述
poolId	LONG	NO	
assetRewards	STRING	NO	
type	INT	NO	0:交易挖矿奖励 , 1:流动性挖矿奖励 , 默认为0
startTime	LONG	NO	
endTime	LONG	NO	
limit	LONG	NO	Default 3, max 100
recvWindow	LONG	NO	
timestamp	LONG	YES	

法币接口

获取法币充值/提现历史记录 (USER_DATA)

响应:

```
{
  "code": "000000",
  "message": "success",
  "data": [
    {
      "orderNo": "7d76d611-0568-4f43-afb6-24cac7767365",
      "fiatCurrency": "BRL", // 法币token
      "indicatedAmount": "10.00", // 交易金额
      "amount": "10.00", // 实际金额(扣除手续费)
      "totalFee": "0.00", // 交易手续费
      "method": "BankAccount", // 交易方式
      "status": "Expired", // Processing, Failed, Successful, Finished, Refunding, Refunded, Refund Failed
      "createTime": 1626144956000, // 订单创建时间
      "updateTime": 1626400907000 // 订单更新时间
    }
  ],
  "total": 1,
  "success": true
}
```

GET /sapi/v1/fiat/orders (HMAC SHA256)

权重(UID): 90000

参数:

名称	类型	是否必需	描述
transactionType	STRING	YES	0-deposit,1-withdraw
beginTime	LONG	NO	
endTime	LONG	NO	
page	INT	NO	默认 1
rows	INT	NO	默认 100, 最大 500
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 若 beginTime 和 endTime 均未发送, 只返回最近30天数据

获取法币支付历史记录 (USER_DATA)

响应:

```
{
  "code": "000000",
  "message": "success",
  "data": [
    {
      "orderNo": "353fca443f06466db0c4dc89f94f027a",
      "sourceAmount": "20.0", // 法币交易数量
      "fiatCurrency": "EUR", // 法币token
      "obtainAmount": "4.462", // 数字货币交易数量
      "cryptoCurrency": "LUNA", // 数字货币token
      "totalFee": "0.2", // 交易手续费
      "price": "4.437472", // 价格
      "status": "Failed", // Processing处理中, Completed完成, Failed失败, Refunded退款
      "paymentMethod": "Credit Card",
      "createTime": 1624529919000, // 订单创建时间
      "updateTime": 1624529919000 // 订单更新时间
    }
  ],
  "total": 1,
  "success": true
}
```

GET /sapi/v1/fiat/payments (HMAC SHA256)

Weight(IP): 1

Parameters:

名称	类型	是否必需	描述
transactionType	STRING	YES	0-buy,1-sell
beginTime	LONG	NO	
endTime	LONG	NO	
page	INT	NO	默认 1
rows	INT	NO	默认 100, 最大 500
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 若beginTime和endTime均未发送,只返回最近30天数据
- paymentMethod : 只有调用购买的历史纪录时 (transactionType=0) , 回传值会有购买方式。目前有四种值 :
 - Cash Balance
 - Credit Card
 - Online Banking
 - Bank Transfer

C2C 接口

获取 C2C 交易历史记录 (USER_DATA)

响应:

```
{
  "code": "000000",
  "message": "success",
  "data": [
    {
      "orderNumber": "20219644646554779648",
      "advNo": "11218246497340923904",
      "tradeType": "SELL",
      "asset": "BUSD",
      "fiat": "CNY",
      "fiatSymbol": "¥",
      "amount": "5000.00000000", // Quantity (in Crypto)
      "totalPrice": "33400.00000000",
      "unitPrice": "6.68", // Unit Price (in Fiat)
      "orderStatus": "COMPLETED", // PENDING, TRADING, BUYER_PAYED, DISTRIBUTING, COMPLETED, IN_APPEAL, CANCELLED
      "createTime": 1619361369000,
      "commission": "0", // Transaction Fee (in Crypto)
      "counterPartNickName": "阿涛✿***",
      "advertisementRole": "TAKER"
    }
  ],
  "total": 1,
  "success": true
}
```

GET /sapi/v1/c2c/orderMatch/listUserOrderHistory (HMAC SHA256)

权重(IP): 1

参数:

名称	类型	是否必需	描述
tradeType	STRING	YES	BUY, SELL
startTimestamp	LONG	NO	
endTimestamp	LONG	NO	
page	INT	NO	default 1
rows	INT	NO	default 100, max 100
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 若startTimestamp和endTimestamp均未发送,只返回最近30天数据。
- startTimestamp和endTimestamp的最大时间间隔为30天。
- 只能查询最近 6 个月的数据。如果需要查看全部C2C订单，你可以前往 <https://c2c.binance.com/zh-CN/fiatOrder>

质押借币接口

获取质押借币资金流水 (USER_DATA)

响应:

```
[
  [
    {
      "asset": "BUSD",
      "type": "borrowIn",
      "amount": "100",
      "timestamp": 1633771139847,
      "tranId": "80423589583"
    },
    [
      {
        "asset": "BUSD",
        "type": "borrowIn",
        "amount": "100",
        "timestamp": 1634638371496,
        "tranId": "81685123491"
      }
    ]
]
```

GET /sapi/v1/loan/income (HMAC SHA256)

权重(UID): 6000

参数:

名称	类型	是否必需	描述
asset	STRING	NO	
type	STRING	NO	默认返回所有类型 枚举值：借入 borrowIn, 抵押金使用 collateralSpent, 还款金额 repayAmount, 抵押物返还 collateralReturn, 增加抵押物 addCollateral, 减少抵押物 removeCollateral, 强平后返还抵押物 collateralReturnAfterLiquidation
startTime	LONG	NO	
endTime	LONG	NO	
limit	INT	NO	默认 20, 最大 100
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 若 startTime 和 endTime 均未发送,只返回最近7天数据
- startTime 和 endTime 的最大时间间隔为30天

借币 - 质押借币借贷 (TRADE)

响应：

```
{
  "loanCoin": "BUSD",
  "loanAmount": "100.5",
  "collateralCoin": "BNB",
  "collateralAmount": "50.5",
  "hourlyInterestRate": "0.001234",
  "orderId": "100000001"
}
```

`POST /sapi/v1/loan/borrow`

权重 (UID)：6000

参数：

名称	类型	是否必需	描述
loanCoin	STRING	YES	
loanAmount	DECIMAL	NO	当collateralAmount为空时，需必填
collateralCoin	STRING	YES	
collateralAmount	DECIMAL	NO	当loanAmount为空时，需必填
loanTerm	INT	YES	7/14/30/90/180 天
recvWindow	LONG	NO	
timestamp	LONG	YES	

借币 - 查询质押借币历史记录 (USER_DATA)

响应：

```
{
  "rows": [
    {
      "orderId": 100000001,
      "loanCoin": "BUSD",
      "initialLoanAmount": "10000",
      "hourlyInterestRate": "0.000057",
      "loanTerm": "7",
      "collateralCoin": "BNB",
      "initialCollateralAmount": "49.27565492",
      "borrowTime": 1575018510000,
      "status": "Repaid" // Accruing_Interest, Overdue, Liquidating, Repaying, Repaid, Liquidated, Pending, F
    }
  ],
}
```

```

    "total": 1
}

```

GET /sapi/v1/loan/borrow/history

权重 (IP) : 400

参数：

名称	类型	是否必需	描述
orderId	LONG	NO	POST /sapi/v1/loan/borrow 中的 orderId
loanCoin	STRING	NO	
collateralCoin	STRING	NO	
startTime	LONG	NO	
endTime	LONG	NO	
current	LONG	NO	当前查询页数，从1开始。默认值：1；最大：1000。
limit	LONG	NO	默认值：10；最大：100。
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 如果没有发送 startTime 和 endTime， 默认返回最近90天的数据。
- startTime 和 endTime 的最大间隔为180天。

借币 - 查询借款中订单列表 (USER_DATA)

响应：

```

{
  "rows": [
    {
      "orderId": 100000001,
      "loanCoin": "BUSD",
      "totalDebt": "10000",
      "residualInterest": "10.27687923",
      "collateralCoin": "BNB",
      "collateralAmount": "49.27565492",
      "currentLTV": "0.57",
      "expirationTime": 1575018510000
    }
  ],
  "total": 1
}

```

GET /sapi/v1/loan/ongoing/orders

权重 (IP) : 400

参数：

名称	类型	是否必需	描述
orderId	LONG	NO	
loanCoin	STRING	NO	
collateralCoin	STRING	NO	
current	LONG	NO	当前查询页数，从1开始。默认值：1；最大：1000。
limit	LONG	NO	默认值：10；最大：100。
recvWindow	LONG	NO	
timestamp	LONG	YES	

还款 - 质押借币还款 (TRADE)

响应：

```
{
  "loanCoin": "BUSD",
  "remainingPrincipal": "100.5",
  "remainingInterest": "0",
  "collateralCoin": "BNB",
  "remainingCollateral": "5.253",
  "currentLTV": "0.25",
  "repayStatus": "Repaid" // Repaid, Repaying
}
```

or

```
{
  "loanCoin": "BUSD",
  "collateralCoin": "BNB",
  "repayStatus": "Repaying" // Repaid, Repaying
}
```

`POST /sapi/v1/loan/repay`

权重 (UID) : 6000

参数：

名称	类型	是否必需	描述
orderId	LONG	YES	
amount	DECIMAL	YES	
type	INT	NO	默认值：1。 1：用借货币还款；2：用抵押币还款。

名称	类型	是否必需	描述
collateralReturn	BOOLEAN	NO	默认值 : TRUE。 TRUE : 多余的抵押金退回现货钱包 ; FALSE: 多余的抵押金保留在原订单里。
recvWindow	LONG	NO	
timestamp	LONG	YES	

还款 - 查询还款记录历史 (USER_DATA)

响应 :

```
{
  "rows": [
    {
      "loanCoin": "BUSD",
      "repayAmount": "10000",
      "collateralCoin": "BNB",
      "collateralUsed": "0"
      "collateralReturn": "49.27565492"
      "repayType": "1" // 1 for "repay with borrowed coin", 2 for "repay with collateral"
      "repayStatus": "Repaid" // Repaid, Repaying, Failed
      "repayTime": 1575018510000
      "orderId": 756783308056935434
    }
  ],
  "total": 1
}
```

GET /sapi/v1/loan/repay/history

权重 (IP) : 400

参数 :

名称	类型	是否必需	描述
orderId	LONG	NO	
loanCoin	STRING	NO	
collateralCoin	STRING	NO	
startTime	LONG	NO	
endTime	LONG	NO	
current	LONG	NO	当前查询页数 , 从1开始。默认值 : 1 ; 最大 : 1000。
limit	LONG	NO	默认值 : 10 ; 最大 : 100。
recvWindow	LONG	NO	

名称	类型	是否必需	描述
timestamp	LONG	YES	

- 如果没有发送startTime和endTime， 默认返回最近90天的数据。
- startTime和endTime的最大间隔为180天。

调整质押率 - 质押借币调整质押率 (TRADE)

响应：

```
{
  "loanCoin": "BUSD",
  "collateralCoin": "BNB",
  "direction": "ADDITIONAL",
  "amount": "5.235",
  "currentLTV": "0.52"
}
```

POST /sapi/v1/loan/adjust/ltv

权重 (UID) : 6000

参数：

名称	类型	是否必需	描述
orderId	LONG	YES	
amount	DECIMAL	YES	
direction	ENUM	YES	"ADDITIONAL", "REDUCED"
recvWindow	LONG	NO	
timestamp	LONG	YES	

调整质押率 - 查询质押率调整历史 (USER_DATA)

响应：

```
{
  "rows": [
    {
      "loanCoin": "BUSD",
      "collateralCoin": "BNB",
      "direction": "ADDITIONAL",
      "amount": "5.235",
      "preLTV": "0.78",
      "afterLTV": "0.56",
      "adjustTime": 1575018510000,
      ...
    }
  ]
}
```

```

    "orderId": 756783308056935434
  }
],
"total": 1
}

```

`GET /sapi/v1/loan/ltv/adjustment/history`

权重 (IP) : 400

参数 :

名称	类型	是否必需	描述
orderId	LONG	NO	
loanCoin	STRING	NO	
collateralCoin	STRING	NO	
startTime	LONG	NO	
endTime	LONG	NO	
current	LONG	NO	当前查询页数，从1开始。默认值：1；最大：1000。
limit	LONG	NO	默认值：10；最大：100。
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 如果没有发送 startTime 和 endTime， 默认返回最近90天的数据。
- startTime 和 endTime 的最大间隔为180天。

查询可借币种数据 (USER_DATA)

响应 :

```
{
  "rows": [
    {
      "loanCoin": "BUSD",
      "_7dHourlyInterestRate": "0.00000491",
      "_7dDailyInterestRate": "0.000118",
      "_14dHourlyInterestRate": "0.00000491",
      "_14dDailyInterestRate": "0.000118",
      "_30dHourlyInterestRate": "0.00000567",
      "_30dDailyInterestRate": "0.000136",
      "_90dHourlyInterestRate": "0.00000596",
      "_90dDailyInterestRate": "0.000143",
      "_180dHourlyInterestRate": "0.00000631",
      "_180dDailyInterestRate": "0.000151",
      "minLimit": "100",
      "maxLimit": "1000000",
      "vipLevel": 1
    }
  ]
}
```

```

    },
],
"total": 1
}

```

`GET /sapi/v1/loan/loanable/data`

获取可借币种的利率和借贷限额。借入限额以美元价值显示。

权重 (IP) : 400

参数 :

名称	类型	是否必需	描述
loanCoin	STRING	NO	
vipLevel	INT	NO	默认 : 用户当前VIP登记。如有特殊配置，则填“-1”查询
recvWindow	LONG	NO	
timestamp	LONG	YES	

查询抵押币种数据 (USER_DATA)

响应 :

```

{
  "rows": [
    {
      "collateralCoin": "BNB",
      "initialLTV": "0.65",
      "marginCallLTV": "0.75",
      "liquidationLTV": "0.83",
      "maxLimit": "10000000",
      "vipLevel": 1
    }
  ],
  "total": 1
}

```

`GET /sapi/v1/loan/collateral/data`

获取抵押币种质押率信息和质押限额。质押限额以美元价值显示。

权重 (IP) : 400

参数 :

名称	类型	是否必需	描述
collateralCoin	STRING	NO	
vipLevel	INT	NO	默认 : 用户当前VIP登记。如有特殊配置，则填“-1”查询

名称	类型	是否必需	描述
recvWindow	LONG	NO	
timestamp	LONG	YES	

查询抵押币种还款汇率 (USER_DATA)

响应：

```
{
  "loanCoin": "BUSD",
  "collateralCoin": "BNB",
  "repayAmount": "1000",
  "rate": "300.36781234" // 抵押/借贷币种的汇率价格
}
```

`GET /sapi/v1/loan/repay/collateral/rate`

获取抵押物还款时，抵押/借贷币种的汇率价格。汇率价格有效时间为8秒。

权重 (IP)：6000

参数：

名称	类型	是否必需	描述
loanCoin	STRING	YES	
collateralCoin	STRING	YES	
repayAmount	DECIMAL	YES	以借贷币种为单位的还款金额
recvWindow	LONG	NO	
timestamp	LONG	YES	

质押借币自定义补仓质押率 (TRADE)

响应：

```
{
  "rows": [
    {
      "orderId": "100000001",
      "collateralCoin": "BNB",
      "preMarginCall": "0.8",
      "afterMarginCall": "0.7",
      "customizeTime": 1575018510000
    }
  ],
}
```

```

    "total": 1
}

```

POST /sapi/v1/loan/customize margin_call

仅可针对进行中订单，自定义补仓质押率。

权重 (UID) : 6000

参数：

名称	类型	是否必需	描述
orderId	LONG	NO	当collateralCoin为空时，需必填。orderId或collateralCoin只需传一个，如果两个参数同时传，以orderId为准。
collateralCoin	STRING	NO	当orderID为空时，需必填。orderId或collateralCoin只需传一个，如果两个参数同时传，以orderId为准。
marginCall	DECIMAL	YES	
recvWindow	LONG	NO	
timestamp	LONG	YES	

Pay 接口

获取 Pay 交易历史记录 (USER_DATA)

响应:

```

{
  "code": "000000",
  "message": "success",
  "data": [
    {
      "orderType": "C2C", //交易类型 枚举值: PAY (C端用户在商户侧消费), PAY_REFUND (C端用户商户侧消费, 退款),
      "transactionId": "M_P_71505104267788288", //流水编号
      "transactionTime": 1610090460133, //交易时间戳
      "amount": "23.72469206", //订单金额 最多8位小数 正为收入, 负为支出
      "currency": "BNB", //订单币种
      "walletType": 1, //1 资金钱包; 2 现货钱包
      "fundsDetail": [ //使用资金明细
        {
          "currency": "USDT", //使用资金币种
          "amount": "1.2" //使用资金金额 为正 最多8位小数
        },
        {
          "currency": "ETH",
          "amount": "0.0001"
        }
      ]
    }
  ]
}

```

```
],
  "success": true
}
```

GET /sapi/v1/pay/transactions (HMAC SHA256)

权重(UID): 3000

参数:

名称	类型	是否必需	描述
startTime	LONG	NO	
endTime	LONG	NO	
limit	INT	NO	默认 100, 最大 100
recvWindow	LONG	NO	
timestamp	LONG	YES	

- 若 startTime 和 endTime 均未发送, 只返回最近90天数据
- startTime 和 endTime 的最大时间间隔为90天
- 支持查询日期范围 : 近18个月以内的订单

闪兑接口

获取闪兑交易记录 (USER_DATA)

响应:

```
{
  "list": [
    {
      "quoteId": "f3b91c525b2644c7bc1e1cd31b6e1aa6",
      "orderId": 940708407462087195, // 订单号
      "orderStatus": "SUCCESS", // 订单状态
      "fromAsset": "USDT", // 闪兑前币种
      "fromAmount": "20", // 闪兑前金额
      "toAsset": "BNB", // 闪兑后币种
      "toAmount": "0.06154036", // 闪兑后金额
      "ratio": "0.00307702", // 价格
      "inverseRatio": "324.99", // 反向价格
      "createTime": 1624248872184
    }
  ],
  "startTime": 1623824139000,
  "endTime": 1626416139000,
  "limit": 100,
  "moreData": false
}
```

GET /sapi/v1/convert/tradeFlow (HMAC SHA256)

权重(UID): 3000**参数:**

名称	类型	是否必需	描述
startTime	LONG	YES	
endTime	LONG	YES	
limit	INT	NO	默认 100, 最大 1000
recvWindow	LONG	NO	
timestamp	LONG	YES	

- startTime和endTime的最大时间间隔为30天

返佣接口

获取现货返佣历史记录 (USER_DATA)

响应:

```
{
  "status": "OK",
  "type": "GENERAL",
  "code": "000000000",
  "data": {
    "page": 1, //当前页
    "totalRecords": 2, //总记录数
    "totalPageNum": 1, //总页数
    "data": [
      {
        "asset": "USDT", // 返佣资产
        "type": 1, // 返佣类型: 1为推荐人返佣, 2为被推荐人返现
        "amount": "0.0001126", // 金额
        "updateTime": 163765132000
      },
      {
        "asset": "ETH",
        "type": 1,
        "amount": "0.00000056",
        "updateTime": 1637928379000
      }
    ]
  }
}
```

GET /sapi/v1/rebate/taxQuery (HMAC SHA256)

权重(UID): 12000**参数:**

名称	类型	是否必需	描述
startTime	LONG	NO	
endTime	LONG	NO	
page	INT	NO	默认 1
recvWindow	LONG	NO	
timestamp	LONG	YES	

- startTime和endTime的最大时间间隔为30天
- 若startTime和endTime均未发送,只返回最近7天数据
- 查询时间最早支持于2020年6月10号

NFT 接口

获取 NFT 资金流水记录 (USER_DATA)

响应:

```
{
  "total": 2, //交易记录总数
  "list": [
    {
      "orderNo": "1_470502070600699904", // 数字前缀含义 0:买单, 1:卖单, 2:版税收入, 3:一级市场买单, 4:mint 费用
      "tokens": [
        {
          "network": "BSC", // NFT的网络
          "tokenId": "216000000496", // NFT的Token ID
          "contractAddress": "MYSTERY_BOX0000087" // NFT 的 Contract Address
        }
      ],
      "tradeTime": 1626941236000, // 交易成功时间
      "tradeAmount": "19.60000000", // 交易金额(实际收入/实际购买总价/实际费用)
      "tradeCurrency": "BNB". // 交易币种
    },
    {
      "orderNo": "1_488306442479116288",
      "tokens": [
        {
          "network": "BSC",
          "tokenId": "132900000007",
          "contractAddress": "0xAf12111a592e408DAbC740849fc5e68629D9fb6"
        }
      ],
      "tradeTime": 1631186130000,
      "tradeAmount": "192.00000000",
      "tradeCurrency": "BNB"
    }
  ]
}
```

```

    }
]
}

```

GET /sapi/v1/nft/history/transactions (HMAC SHA256)

权重(UID): 3000

参数:

名称	类型	是否必需	描述
orderType	INT	YES	0:买单，1:卖单，2:版税收入，3:一级市场买单，4:mint 费用
startTime	LONG	NO	
endTime	LONG	NO	
limit	INT	NO	默认 50, 最大 50
page	INT	NO	默认 1
recvWindow	LONG	NO	
timestamp	LONG	YES	

- startTime和endTime的最大时间间隔为90天
- 若startTime和endTime均未发送,只返回最近7天数据

获取 NFT 充值记录 (USER_DATA)

响应:

```
{
  "total": 2,
  "list": [
    {
      "network": "ETH", // NFT的网络
      "txID": null, // 该笔充值记录的 Transaction ID
      "contractAddress": "0xe507c961ee127d4439977a61af39c34eafee0dc6", // NFT的 Contract Address
      "tokenId": "10014", // NFT的 Token ID
      "timestamp": 1629986047000
    },
    {
      "network": "BSC",
      "txID": null,
      "contractAddress": "0x058451b463bab04f52c0799d55c4094f507acfa9",
      "tokenId": "10016",
      "timestamp": 1630083581000
    }
  ]
}
```

GET /sapi/v1/nft/history/deposit (HMAC SHA256)

权重(UID): 3000**参数:**

名称	类型	是否必需	描述
startTime	LONG	NO	
endTime	LONG	NO	
limit	INT	NO	默认 50, 最大 50
page	INT	NO	默认 1
recvWindow	LONG	NO	
timestamp	LONG	YES	

- startTime和endTime的最大时间间隔为90天
- 若startTime和endTime均未发送,只返回最近7天数据

获取 NFT 提现记录 (USER_DATA)

响应:

```
{
  "total": 178,
  "list": [
    {
      "network": "ETH",
      "txID": "0x2be5eed31d787fdb4880bc631c8e76bdfb6150e137f5cf1732e0416ea206f57f",
      "contractAddress": "0xe507c961ee127d4439977a61af39c34eafee0dc6", // NFT的 Contract Address
      "tokenId": "1000001247", // NFT的 Token ID
      "timestamp": 1633674433000, // 提现时间
      "fee": 0.1, // 提现手续费
      "feeAsset": "ETH" // 手续费币种
    },
    {
      "network": "ETH",
      "txID": "0x3b3aea5c0a4faccd6f306641e6deb9713ab229ac233be3be227f580311e4362a",
      "contractAddress": "0xe507c961ee127d4439977a61af39c34eafee0dc6",
      "tokenId": "40000030",
      "timestamp": 1633677022000,
      "fee": 0.1,
      "feeAsset": "ETH"
    }
  ]
}
```

GET /sapi/v1/nft/history/withdraw (HMAC SHA256)

权重(UID): 3000**参数:**

名称	类型	是否必需	描述
startTime	LONG	NO	
endTime	LONG	NO	
limit	INT	NO	默认 50, 最大 50
page	INT	NO	默认 1
recvWindow	LONG	NO	
timestamp	LONG	YES	

- startTime和endTime的最大时间间隔为90天
- 若startTime和endTime均未发送,只返回最近7天数据

获取 NFT 资产 (USER_DATA)

响应:

```
{
  "total": 347,
  "list": [
    {
      "network": "BSC", // NFT的网络
      "contractAddress": "REGULAR11234567891779", // NFT的 Contract Address
      "tokenId": "100900000017" // NFT的 Token ID
    },
    {
      "network": "BSC",
      "contractAddress": "SSMDQ8W59",
      "tokenId": "200500000011"
    },
    {
      "network": "BSC",
      "contractAddress": "SSMDQ8W59",
      "tokenId": "200500000019"
    }
  ]
}
```

GET /sapi/v1/nft/user/getAsset (HMAC SHA256)

权重(UID): 3000

参数:

名称	类型	是否必需	描述
limit	INT	NO	默认 50, 最大 50
page	INT	NO	默认 1
recvWindow	LONG	NO	

名称	类型	是否必需	描述
timestamp	LONG	YES	

币安码接口

币安码为一串预先充值的密码，每串密码承载加密货币的价值。通过币安码解决方案可进行简易，快速，安全的加密资产的交易流通。币安码API旨在促进币安码的即时创建、兑现和价值验证。币安码由两部分组成：参考号和币安码。参考号可公开流通，可用于验证币安码的有效性；币安码应妥善保管，因为只要有人拥有该码，就可以随时兑现。

请注意，以下接口暂不支持子账户使用

创建币安码 (USER_DATA)

响应:

```
{
  "code": "000000",
  "message": "success",
  "data": {
    "referenceNo": "0033002327977405", //参考号
    "code": "AOGANK3NB4GIT3C6" //币安码
  },
  "success": true
}
```

POST /sapi/v1/giftcard/createCode (HMAC SHA256)

该API用于创建一个币安码。要开始使用，请确保：

- 你有一个币安账户
- 你已通过了 KYC
- 币安资金账户中有足够的余额
- 你的API Key需要开启允许提现权限

权重(IP): 1

每日制码金额上限：2 BTC / 24H

每日制码次数上限：200 次 / 24H

参数:

名称	类型	是否必需	描述
token	STRING	YES	币安码中的数字货币币种
amount	DOUBLE	YES	币安码中的数字货币数量
recvWindow	LONG	NO	
timestamp	LONG	YES	

兑现币安码 (USER_DATA)

响应:

```
{
  "code": "000000",
  "message": "success",
  "data": {
    "token": "BNB", //币种
    "amount": "10", //金额
    "referenceNo": "0033002327977405", //参考号
    "identityNo": "10316281761814589440" //无意义, 请忽略
  },
  "success": true
}
```

`POST /sapi/v1/giftcard/redeemCode (HMAC SHA256)`

该API用于兑现币安码，兑现后币种将存入您的资金账户

请注意，如果您在 24 小时内输入错误币安码 5 次，您将无法在当天兑现任何币安码

权重(IP): 1

参数:

名称	类型	是否必需	描述
code	STRING	YES	用于赎回的binance code，支持加密&未加密两种方式
externalUid	String	NO	每个外部用户 ID 代表合作伙伴平台上的某个用户。该功能帮助您识别不同用户的兑现行为，例如兑现频次和金额。它还有助于对单个账户进行风险和限额控制，例如设置单个账户每日兑现金额、频次和卡密输错次数的上限。这也将防止单个账户突破合作伙伴的每日兑现限额从而导致合作伙伴的账户在当日无法继续制码或者兑现。如果您有外部的网站且有不同的用户在您的平台上兑现 Binance Code 或礼品卡，我们强烈建议您使用此功能并将您用户的用户 ID 传输给我们来进行风控。为保护用户的信息安全，您可以选择以任何格式（上限为 400 个字符）传输用户 ID。
recvWindow	LONG	NO	
timestamp	LONG	YES	

注意：

- 参数code有两种形式传输
 - Plaintext: 未加密的格式
 - Encrypted: 加密后的格式
- 用加密后的格式传输更安全，传输加密的格需要以下步骤：
 - 调用获取公钥的API
 - 用加密算法对公钥和原始的binance code进行加密：`RSA/ECB/OAEPWithSHA-256AndMGF1Padding`

获取加密后的binance code的java使用方式:

```

private static PublicKey getPublicKey(String publicKey) throws Exception {
    KeyFactory keyFactory = KeyFactory.getInstance("RSA");
    byte[] decodedKey = Base64.decodeBase64(publicKey.getBytes());
    X509EncodedKeySpec keySpec = new X509EncodedKeySpec(decodedKey);
    return keyFactory.generatePublic(keySpec);
}

public static String encrypt(String content, String publicKeyString) throws Exception {
    if (StringUtils.isEmpty(new CharSequence[]{content, publicKeyString})) {
        throw new IllegalArgumentException("invalid content or privateKey.");
    } else {
        Cipher cipher = Cipher.getInstance("RSA/ECB/OAEPWITHSHA-256ANDMGF1PADDING", "BC");
        cipher.init(Cipher.ENCRYPT_MODE, getPublicKey(publicKeyString));
        return new String(Base64.encodeBase64URLSafe(cipher.doFinal(content.getBytes("UTF-8")))));
    }
}

static {
    Security.addProvider(new BouncyCastleProvider());
}

```

验证币安码 (USER_DATA)

响应:

```
{
    "code": "000000",
    "message": "success",
    "data": {
        "valid": true, //是否有效
        "token": "BNB", //币种
        "amount": "0.00000001" //金额
    },
    "success": true
}
```

GET /sapi/v1/giftcard/verify (HMAC SHA256)

此 API 用于通过输入参考号来验证币安码是否有效

请注意，如果您在一小时内输入错误的币安码 5 次，您将无法在该小时内验证任何币安码

权重(IP): 1

参数:

名称	类型	是否必需	描述
referenceNo	STRING	YES	参考号
recvWindow	LONG	NO	
timestamp	LONG	YES	

获取RSA Public Key (USER_DATA)

响应:

```
{
  "code": "000000",
  "message": "success",
  "data": "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCXBBVLAc1GQ5FsIFFqOHRPTox5noBONIKr+IAedTR9FkVxq6e65updE",
  "success": true
}
```

GET /sapi/v1/giftcard/cryptography/rsa-public-key (HMAC SHA256)

此API用来获取用户的公钥

公钥可以用来对code进行加密

请注意公钥获取只有当天有效

权重(IP): 1

参数:

名称	类型	是否必需	描述
recvWindow	LONG	NO	
timestamp	LONG	YES	

购买币安码 (TRADE)

响应 :

```
{
  "code": "000000",
  "message": "success",
  "data": {
    "referenceNo": "0033002327977405",
    "code": "AOGANK3NB4GIT3C6"
  },
  "success": true
}
```

POST /sapi/v1/giftcard/buyCode (HMAC SHA256)

- 该API用于购买一个币安码。
- 你可以透过支持的数字货币来购买不同面额的币安码。
- 在购买后相应数量的数字货币将从你的资金账户被扣除。
- 要开始使用，请确保：
 - 你有一个币安账户

- 你已通过了 KYC
- 币安资金账户中有足够的余额
- 你的API Key需要开启允许提现权限

权重(IP): 1

- 每日制码金额上限： 2 BTC / 24H
- 每日制码次数上限： 200 次 / 24H

参数：

名称	类型	是否必需	描述
baseToken	STRING	YES	你用来支付的货币，例如：BUSD
faceToken	STRING	YES	你购买的礼品卡面额，例如：BNB。如果 faceToken = baseToken，将等同于使用 createCode API
baseTokenAmount	DOUBLE	YES	支付的货币数量，例如：1.002
recvWindow	LONG	NO	
timestamp	LONG	YES	

获取货币使用限制 (USER_DATA)

响应：

```
{
  "code": "000000",
  "message": "success",
  "data": [
    {
      "coin": "BNB",
      "fromMin": "0.01",
      "fromMax": "1"
    }
  ],
  "success": true
}
```

`GET /sapi/v1/giftcard/buyCode/token-limit (HMAC SHA256)`

此 API 是用来查看你所支付的数字货币，可以购买的面额与数量限制。

权重(IP): 1**参数：**

名称	类型	是否必需	描述
baseToken	STRING	YES	你用来支付的货币，例如：BUSD
recvWindow	LONG	NO	
timestamp	LONG	YES	

错误代码

错误JSON格式:

```
{  
  "code": -1121,  
  "msg": "Invalid symbol."  
}
```

错误由两部分组成：错误代码和消息。 代码是通用的，但是消息可能会有所不同。

10xx -常规服务器或网络问题

-1000 UNKNOWN

- 处理请求时发生未知错误。
- 处理请求时发生未知错误。 [%s]

-1001 DISCONNECTED

- 内部错误; 无法处理您的请求。 请再试一次.

-1002 UNAUTHORIZED

- 您无权执行此请求。

-1003 TOO_MANY_REQUESTS

- 排队的请求过多。
- 请求权重过多； 请使用websocket获取实时更新。
- 请求权重过多； 当前限制为每分钟 %s 请求权重。 请使用websocket进行实时更新，以避免轮询API。
- 请求权重过多； IP被禁止，直到 %s。 请使用websocket进行实时更新，以免被禁。

-1004 SERVER_BUSY

- 服务器正忙，请稍候再试。

-1006 UNEXPECTED_RESP

- 从消息总线收到意外的响应。 执行状态未知。

-1007 TIMEOUT

- 等待后端服务器响应超时。 发送状态未知； 执行状态未知。

-1008 SERVER_BUSY

- 现货交易服务器当前因其他请求而过载。 请在几分钟后重试。

-1014 UNKNOWN_ORDER_COMPOSITION

- 不支持的订单组合。

-1015 TOO_MANY_ORDERS

- 新订单太多。
- 新订单太多； 当前限制为每 %s %s 个订单。

-1016 SERVICE_SHUTTING_DOWN

- 该服务不可用。

-1020 UNSUPPORTED_OPERATION

- 不支持此操作。

-1021 INVALID_TIMESTAMP

- 此请求的时间戳在recvWindow之外。
- 此请求的时间戳比服务器时间提前1000毫秒。

-1022 INVALID_SIGNATURE

- 此请求的签名无效。

-1099 NOT FOUND, AUTHENTICATED, OR AUTHORIZED

- 替换错误代码-1999

11xx - 2xxx Request issues**-1100 ILLEGAL_CHARS**

- 在参数中发现非法字符。
- 在参数中发现非法字符。%s
- 在参数%s中发现非法字符；合法范围是%s。

-1101 TOO_MANY_PARAMETERS

- 为此端点发送的参数太多。
- 参数太多；预期为%s并收到了%s。
- 检测到的参数值重复。

-1102 MANDATORY_PARAM_EMPTY_OR_MALFORMED

- 未发送强制性参数，该参数为空/空或格式错误。
- 强制参数%s未发送，为空/空或格式错误。
- 必须发送参数%s或%s，但两者均为空！

-1103 UNKNOWN_PARAM

- 发送了未知参数。

-1104 UNREAD_PARAMETERS

- 并非所有发送的参数都被读取。
- 并非所有发送的参数都被读取；读取了%s参数，但被发送了%s。

-1105 PARAM_EMPTY

- 参数为空。
- 参数%s为空。

-1106 PARAM_NOT_REQUIRED

- 不需要时已发送参数。
- 不需要时发送参数%s。

-1111 BAD_PRECISION

- 精度超过为此资产定义的最大值。

-1112 NO_DEPTH

- 交易对没有挂单。

-1114 TIF_NOT_REQUIRED

- 不需要时发送了TimeInForce参数。

-1115 INVALID_TIF

- 无效 timeInForce.

-1116 INVALID_ORDER_TYPE

- 无效订单类型。

-1117 INVALID_SIDE

- 无效买卖方向。

-1118 EMPTY_NEW_CL_ORD_ID

- 新的客户订单ID为空。

-1119 EMPTY_ORG_CL_ORD_ID

- 客户自定义的订单ID为空。

-1120 BAD_INTERVAL

- 无效时间间隔。

-1121 BAD_SYMBOL

- 无效的交易对。

-1125 INVALID_LISTEN_KEY

- 该listenKey不存在。

-1127 MORE_THAN_XX_HOURS

- 查询间隔太大。
- 从开始时间到结束时间之间超过% s 小时。

-1128 OPTIONAL_PARAMS_BAD_COMBO

- 可选参数组合无效。

-1130 INVALID_PARAMETER

- 发送的参数为无效数据。
- 发送参数% s 的数据无效。

-1131 BAD_RECV_WINDOW

- recvWindow 必须小于 60000

-1134 BAD_STRATEGY_TYPE

- strategyType 必须小于 1000000

-2010 NEW_ORDER_REJECTED

- 新订单被拒绝

-2011 CANCEL_REJECTED

- 取消订单被拒绝

-2013 NO_SUCH_ORDER

- 订单不存在。

-2014 BAD_API_KEY_FMT

- API-key 格式无效。

-2015 REJECTED_MBX_KEY

- 无效的API密钥，IP或操作权限。

-2016 NO_TRADING_WINDOW

- 找不到该交易对的交易窗口。尝试改为24小时自动报价。

3xxx-5xxx SAPI 具体问题

-3000 INNER_FAILURE

- 内部服务器错误。

-3001 NEED_ENABLE_2FA

- 请先启用2FA。

-3002 ASSET_DEFICIENCY

- 此资产不存在。

-3003 NO_OPENED_MARGIN_ACCOUNT

- 杠杆账户不存在。

-3004 TRADE_NOT_ALLOWED

- 禁止交易。

-3005 TRANSFER_OUT_NOT_ALLOWED

- 不允许转账。

-3006 EXCEED_MAX_BORROWABLE

- 您的已借金额已超过最高可借金额。

-3007 HAS_PENDING_TRANSACTION

- 您有待处理的交易，请稍后再试。

-3008 BORROW_NOT_ALLOWED

- 不允许借款。

-3009 ASSET_NOT_MORTGAGEABLE

- 此资产目前不允许转入杠杆账户。

-3010 REPAY_NOT_ALLOWED

- 不允许还款。

-3011 BAD_DATE_RANGE

- 您输入的日期无效。

-3012 ASSET_ADMIN_BAN_BORROW

- 此资产禁止借款。

-3013 LT_MIN_BORROWABLE

- 借入金额少于最低借入金额。

-3014 ACCOUNT_BAN_BORROW

- 此帐户禁止借款。

-3015 REPAY_EXCEED LIABILITY

- 还款额超过借款额。

-3016 LT_MIN_REPAY

- 还款额少于最低还款额。

-3017 ASSET_ADMIN_BAN_MORTGAGE

- 此资产目前不允许转入保证金账户。

-3018 ACCOUNT_BAN_MORTGAGE

- 此帐户已禁止转入。

-3019 ACCOUNT_BAN_ROLLOUT

- 此帐户禁止转出。

-3020 EXCEED_MAX_ROLLOUT

- 转出金额超过上限。

-3021 PAIR_ADMIN_BAN_TRADE

- 杠杆账户无法交易此交易对。

-3022 ACCOUNT_BAN_TRADE

- 账号被禁止交易。

-3023 WARNING_MARGIN_LEVEL

- 无法在当前杠杆倍数下转出资金或者下单

-3024 FEW LIABILITY_LEFT

- 付款之后未付款的债务太小

-3025 INVALID_EFFECTIVE_TIME

- 输入时间有误。

-3026 VALIDATION_FAILED

- 输入参数有误。

-3027 NOT_VALID_MARGIN_ASSET

- 无效的杠杆资产。

-3028 NOT_VALID_MARGIN_PAIR

- 无效的杠杆交易对。

-3029 TRANSFER_FAILED

- 转账失败。

-3036 ACCOUNT_BAN_REPAY

- 此账号无法还款。

-3037 PNL_CLEARING

- PNL正在清帐，请稍等。

-3038 LISTEN_KEY_NOT_FOUND

- 找不到Listen key

-3041 BALANCE_NOT_CLEARED

- 余额不足

-3042 PRICE_INDEX_NOT_FOUND

- 该杠杆交易对无可用价格指数。

-3043 TRANSFER_IN_NOT_ALLOWED

- 不允许转入。

-3044 SYSTEM_BUSY

- 系统繁忙。

-3045 SYSTEM

- 系统目前没有足够可借的资产。

-3999 NOT_WHITELIST_USER

- 此功能只面向邀请的用户。

-4001 CAPITAL_INVALID

- 非法操作

-4002 CAPITAL_IG

- 非法获取

-4003 CAPITAL_IEV

- 非法邮箱验证

-4004 CAPITAL_UA

- 未登录或者认证。

-4005 CAPITAL_TOO_MANY_REQUEST

- 请求太频繁。

-4006 CAPITAL_ONLY_SUPPORT_PRIMARY_ACCOUNT

- 只支持主账号。

-4007 CAPITAL_ADDRESS_VERIFICATION_NOT_PASS

- 地址的没有通过校验。

-4008 CAPITAL_ADDRESS_TAG_VERIFICATION_NOT_PASS

- 地址的标记信息(tag)没有通过校验。

-4010 CAPITAL_WHITELIST_EMAIL_CONFIRM

- 确认电子邮件已经列入白名单。

-4011 CAPITAL_WHITELIST_EMAIL_EXPIRED

- 列入白名单的电子邮件无效。

-4012 CAPITAL_WHITELIST_CLOSE

- 白名单未打开。

-4013 CAPITAL_WITHDRAW_2FA_VERIFY

- 2FA未打开。

-4014 CAPITAL_WITHDRAW_LOGIN_DELAY

- 在登录后的2分钟之内不允许提款。

-4015 CAPITAL_WITHDRAW_RESTRICTED_MINUTE

- 暂停提款

-4016 CAPITAL_WITHDRAW_RESTRICTED_PASSWORD

- 在密码修改后的24小时之内不允许提款。

-4017 CAPITAL_WITHDRAW_RESTRICTED_UNBIND_2FA

- 在2FA发行后的24小时之内不允许提款。

-4018 CAPITAL_WITHDRAW_ASSET_NOT_EXIST

- 此资产不存在。

-4019 CAPITAL_WITHDRAW_ASSET_PROHIBIT

- 此资产不允许提款。

-4021 CAPITAL_WITHDRAW_AMOUNT_MULTIPLE

- 资产的提款数量必须是%s的%s倍。

-4022 CAPITAL_WITHDRAW_MIN_AMOUNT

- 不须少于最低的提款数量%s。

-4023 CAPITAL_WITHDRAW_MAX_AMOUNT

- 在24小时之内，不须超过最高的提款数量。

-4024 CAPITAL_WITHDRAW_USER_NO_ASSET

- 当前用户没有此资产。

-4025 CAPITAL_WITHDRAW_USER_ASSET_LESS_THAN_ZERO

- 持有资产的数量小于零。

-4026 CAPITAL_WITHDRAW_USER_ASSET_NOT_ENOUGH

- 此资产余额不足。

-4027 CAPITAL_WITHDRAW_GET_TRAN_ID_FAILURE

- 无法获取tranId。

-4028 CAPITAL_WITHDRAW_MORE_THAN_FEE

- 提款金额必须多于佣金金额。

-4029 CAPITAL_WITHDRAW_NOT_EXIST

- 此提款记录不存在。

-4030 CAPITAL_WITHDRAW_CONFIRM_SUCCESS

- 提款资产成功。

-4031 CAPITAL_WITHDRAW_CANCEL_FAILURE

- 取消提款失败。

-4032 CAPITAL_WITHDRAW_CHECKSUM_VERIFY_FAILURE

- 验证提款失败。

-4033 CAPITAL_WITHDRAW_ILLEGAL_ADDRESS

- 提款地址不合法。

-4034 CAPITAL_WITHDRAW_ADDRESS_CHEAT

- 当前地址有异常。

-4035 CAPITAL_WITHDRAW_NOT_WHITE_ADDRESS

- 此地址不在白名单上。请加入然后重试。

-4036 CAPITAL_WITHDRAW_NEW_ADDRESS

- 新地址在{0}小时后才可以提款。

-4037 CAPITAL_WITHDRAW_RESEND_EMAIL_FAIL

- 重新发送邮件失败。

-4038 CAPITAL_WITHDRAW_RESEND_EMAIL_TIME_OUT

- 请5分钟后重试。

-4039 CAPITAL_USER_EMPTY

- 用户不存在。

-4041 CAPITAL_MINUTE_TOO_SMALL

- 请一分钟后重试。

-4042 CAPITAL_CHARGE_NOT_RESET

- 资产无法重新获取存款地址。

-4043 CAPITAL_ADDRESS_TOO MUCH

- 在24小时之内充值超过100多个地址。

-4044 CAPITAL_BLACKLIST_COUNTRY_GET_ADDRESS

- 此国家在黑名单上。

-4045 CAPITAL_GET_ASSET_ERROR

- 获得资产失败。

-4046 CAPITAL AGREEMENT NOT CONFIRMED

- 协议未确认。

-4047 CAPITAL_DATE_INTERVAL_LIMIT

- 时间间隔必须在0-90天之内

-5001 ASSET_DRIBBLET_CONVERT_SWITCH_OFF

- 不允许转移到微型资产。

-5002 ASSET_ASSET_NOT_ENOUGH

- 此余额不足。

-5003 ASSET_USER_HAVE_NO_ASSET

- 此资产不存在。

-5004 USER_OUT_OF_TRANSFER_FLOAT

- 剩余余额已超过0.001BTC，请重新选择。
- %s的剩余余额已超过0.001BTC，请重新选择。

-5005 USER_ASSET_AMOUNT_IS_TOO_LOW

- BTC的剩余余额太低，请重新选择。
- %s的剩余余额太低，请重新选择。

-5006 USER_CAN_NOT_REQUEST_IN_24_HOURS

- 24小时内只能转账一次。

-5007 AMOUNT_OVER_ZERO

- 数量必须大于零。

-5008 ASSET_WITHDRAW_WITHDRAWING_NOT_ENOUGH

- 可退回资产的金额不足。

-5009 PRODUCT_NOT_EXIST

- 产品不存在。

-5010 TRANSFER_FAIL

- 资产转移失败。

-5011 FUTURE_ACCT_NOT_EXIST

- 合约帐户不存在。

-5012 TRANSFER_PENDING

- 资产转移正在进行中。

-5021 PARENT_SUB_HAVE_NO_RELATION

- 当前的子账户和母账户没有从属关系。

-5012 FUTURE_ACCT_OR_SUBRELATION_NOT_EXIST

- 合约帐户或子账户关系不存在。

6XXX - 币安宝相关

-6001 DAILY_PRODUCT_NOT_EXIST

- 理财产品不存在。

-6003 DAILY_PRODUCT_NOT_ACCESSIBLE

- 产品不存在或者没有权限。

-6004 DAILY_PRODUCT_NOT_PURCHASABLE

- 产品无法购买。

-6005 DAILY_LOWER_THAN_MIN_PURCHASE_LIMIT

- 低于可以购买的最小限额。

-6006 DAILY_REDEEM_AMOUNT_ERROR

- 赎回额度有误。

-6007 DAILY_REDEEM_TIME_ERROR

- 不在赎回的时间内。

-6008 DAILY_PRODUCT_NOT_REDEEMABLE

- 产品暂时无法赎回。

-6009 REQUEST_FREQUENCY_TOO_HIGH

- 发送请求太频繁。

-6011 EXCEEDED_USER_PURCHASE_LIMIT

- 超购每个月用户可以申购的最大次数。

-6012 BALANCE_NOT_ENOUGH

- 余额不足。

-6013 PURCHASING_FAILED

- 申购失败。

-6014 UPDATE_FAILED

- 超过可以申购的最大上限。

-6015 EMPTY_REQUEST_BODY

- 请求的body为空。

-6016 PARAMS_ERR

- 请求的参数有误。

-6017 NOT_IN_WHITELIST

- 不在白名单里面。

-6018 ASSET_NOT_ENOUGH

- 资产不足。

-6019 PENDING

- 需要进一步确认。

-6020 PROJECT_NOT_EXISTS

- 此项目不存在。

70xx - 期货

-7001 FUTURES_BAD_DATE_RANGE

- 此日期范围不支持。

-7002 FUTURES_BAD_TYPE

- 此数据请求类型不支持。

20xxx - 期货策略交易

-20121 INVALID SYMBOL

- 无效交易对。

-20124 INVALID ALGO ID OR IT HAS BEEN COMPLETED

- 无效的策略订单ID或者它已经被执行。

-20130 INVALID DATA SENT FOR A PARAMETER

- 无效数据。

-20132 THE CLIENT ALGO ID IS DUPLICATED

- 用户自定义策略订单ID重复。

-20194 DURATION IS TOO SHORT TO EXECUTE ALL REQUIRED QUANTITY

- Duration 时间太短不足以执行用户选择的订单数量。

-20195 THE TOTAL SIZE IS TOO SMALL

- 下单数量太小。

-20196 THE TOTAL SIZE IS TOO LARGE

- 下单数量太大。

-20198 REACH THE MAX OPEN ORDERS ALLOWED

- 达到了最大挂单上限。

-9xxx 过滤器故障

报错信息	描述
"Filter failure: PRICE_FILTER"	"价格"过高，过低和/或不遵循交易对的最小价格规则。
"Filter failure: PERCENT_PRICE"	"价格"比最近Y分钟的平均加权价格高X%或X%太低。
"Filter failure: PERCENT_PRICE_BY_SIDE"	<code>price</code> 在当前方向上(<code>BUY</code> 或者 <code>SELL</code>)比 <code>lastPrice</code> 价格超过X%或者低于Y%。
"Filter failure: LOT_SIZE"	"数量"太高，太低和/或不遵循该交易对的步长规则。
"Filter failure: MIN_NOTIONAL"	价格*数量太低，无法成为该交易对的有效订单。
"Filter failure: ICEBERG_PARTS"	<code>ICEBERG</code> 订单会分成太多部分；icebergQty太小。
"Filter failure: MARKET_LOT_SIZE"	"MARKET"订单的"数量"过高，过低和/或未遵循交易对的步长规则。
"Filter failure: MAX_POSITION"	达到账户的最大仓位限制。这包括了账户的余额总额，以及所有处于open的买单的数量总和。
"Filter failure: MAX_NUM_ORDERS"	客户在交易对上有太多挂单。
"Filter failure: MAX_ALGO_ORDERS"	账户有太多未平仓止损和/或在交易对上执行获利指令。

报错信息	描述
"Filter failure: MAX_NUM_ICEBERG_ORDERS"	客户在交易对上有太多 iceberg 挂单。
"Filter failure: TRAILING_DELTA"	<code>trailingDelta</code> 值不在限定的范围内。
"Filter failure: EXCHANGE_MAX_NUM_ORDERS"	帐户上的交易所有太多挂单。
"Filter failure: EXCHANGE_MAX_ALGO_ORDERS"	帐户有太多止损挂单和/或在交易所收取获利指令。

10xxx - 质押借币

-10001 SYSTEM_MAINTENANCE

- 系统维护中，请稍后再试

-10002 INVALID_INPUT

- 无效的输入参数

-10005 NO_RECORDS

- 暂无记录

-10007 COIN_NOT_LOANABLE

- 该币种暂不支持借贷

-10008 COIN_NOT_LOANABLE

- 该币种暂不支持借贷

-10009 COIN_NOT_COLLATERAL

- 该币种暂不支持抵押

-10010 COIN_NOT_COLLATERAL

- 该币种暂不支持抵押

-10011 INSUFFICIENT_ASSET

- 现货资产不足

-10012 INVALID_AMOUNT

- 无效的还款金额

-10013 INSUFFICIENT_AMOUNT

- 抵押资产不足

-10015 DEDUCTION_FAILED

- 抵押资产扣款失败

-10016 LOAN_FAILED

- 放贷失败

-10017 REPAY_EXCEED_DEBT

- 还款金额超过负债金额

-10018 INVALID_AMOUNT

- 无效的还款金额

-10019 CONFIG_NOT_EXIST

- 配置不存在

-10020 UID_NOT_EXIST

- 用户ID不存在

-10021 ORDER_NOT_EXIST

- 订单不存在

-10022 INVALID_AMOUNT

- 无效的调整金额

-10023 ADJUST_LTV_FAILED

- 调整质押率失败

-10024 ADJUST_LTV_NOT_SUPPORTED

- 暂不支持调整质押率

-10025 REPAY_FAILED

- 还款失败

-10026 INVALID_PARAMETER

- 无效的参数

-10028 INVALID_PARAMETER

- 无效的参数

-10029 AMOUNT_TOO_SMALL

- 借贷金额过小

-10030 AMOUNT_TOO_LARGE

- 借贷金额过大

-10031 QUOTA_REACHED

- 已达到个人借贷限额

-10032 REPAY_NOT_AVAILABLE

- 暂不支持换款

-10034 REPAY_NOT_AVAILABLE

- 抵押物还款暂时不支持，请尝试用借货币还款。

-10039 AMOUNT_TOO_SMALL

- 还款金额过小

-10040 AMOUNT_TOO_LARGE

- 还款金额过大

-10041 INSUFFICIENT_AMOUNT

- 由于借贷需求过多，系统剩余可借{0}额度不足。请调整借贷金额或明天再试。

-10042 ASSET_NOT_SUPPORTED

- 暂不支持%s币种

-10043 ASSET_NOT_SUPPORTED

- 暂不支持{0}借贷

-10044 QUOTA_REACHED

- 抵押物数量已达到限额，请调整抵押金额或使用其他抵押资产。

-10045 COLLTERAL_REPAY_NOT_SUPPORTED

- 该借贷币种暂不支持抵押物还款，请稍后再试。

-10046 EXCEED_MAX_ADJUSTMENT

- 调整抵押物超过最大限额，请重试。

-10047 REGION_NOT_SUPPORTED

- 受当地法规管制，您所在地区暂不支持该币种。

13xxx - 杠杆代币

-13000 BLVT_FORBID_REDEEM

- 当前该杠杆代币关闭赎回

-13001 BLVT_EXCEED_DAILY_LIMIT

- 超过该代币个人24小时赎回金额上限

-13002 BLVT_EXCEED_TOKEN_DAILY_LIMIT

- 超过该代币全局24小时赎回金额上限

-13003 BLVT_FORBID_PURCHASE

- 当前该杠杆代币关闭申购

-13004 BLVT_EXCEED_DAILY_PURCHASE_LIMIT

- 超过该代币个人24小时申购金额上限

-13005 BLVT_EXCEED_TOKEN_DAILY_PURCHASE_LIMIT

- 超过该代币全局24小时申购金额上限

-13006 BLVT_PURCHASE_LESS_MIN_AMOUNT

- 申购金额低于规定下限

-13007 BLVT_PURCHASE AGREEMENT NOT SIGN

- 没有签署开通交易协议

12xxx - 流动性挖矿

-12014 TOO MANY REQUESTS

- 2秒内接收的请求数量多于1条

18xxx - 币安码**-18002**

- The total amount of codes you created has exceeded the 24-hour limit, please try again after UTC 0
- 24小时内制码总金额已超过限额，请UTC0点后再尝试

-18003

- Too many codes created in 24 hours, please try again after UTC 0
- 24小时内制码总次数已超过限额，请UTC0点后再尝试

-18004

- Too many invalid redeem attempts in 24 hours, please try again after UTC 0
- 24小时内兑币安码输错次数已超过限额，请UTC0点后再尝试

-18005

- Too many invalid verify attempts, please try later
- 参考号输错次数过多，请稍后再试

-18006

- The amount is too small, please re-enter
- 金额过小，请重新输入

-18007

- This token is not currently supported, please re-enter
- 尚未支持该币种，请重新输入

21xxx - 统一帐户**-21001 USER_IS_NOT_UNIACCOUNT**

- 尚未开通统一账户。

-21002 UNI_ACCOUNT_CANT_TRANSFER_FUTURE

- 统一账户禁用margin向futures转账。

-21003 NET_ASSET_MUST_LTE_RATIO

- margin资产更新失败。

-21004 USER_NO LIABILITY

- 用户不存在统一账户穿仓负债

-21005 NO_ENOUGH_ASSET

- 用户现货钱包BUSD资产不足以偿还统一账户穿仓负债

-21006 HAD_IN_PROCESS_REPAY

- 用户存在正在偿还的统一账户穿仓负债

-21007 IN_FORCE_LIQUIDATION

- 强平进行中，用户偿还统一账户穿仓负债失败

订单拒绝错误

以下错误代码表示撮合引擎返回的订单相关错误:

- 1010 ERROR_MSG RECEIVED
- 2010 NEW_ORDER_REJECTED
- 2011 CANCEL_REJECTED

结合以下消息将指示特定的错误：

Error message	Description
"Unknown order sent."	找不到订单(通过"orderId" , "clientOrderId" , "origClientId")
"Duplicate order sent."	<code>clientOrderId</code> 已经被使用
"Market is closed."	该交易对不在交易范围
"Account has insufficient balance for requested action."	没有足够的资金来完成行动
"Market orders are not supported for this symbol."	交易对上未启用"MARKET"
"Iceberg orders are not supported for this symbol."	交易对上未启用 <code>icebergQty</code>
"Stop loss orders are not supported for this symbol."	交易对上未启用 <code>STOP_LOSS</code>
"Stop loss limit orders are not supported for this symbol."	交易对上未启用 <code>STOP_LOSS_LIMIT</code>
"Take profit orders are not supported for this symbol."	交易对上未启用 <code>TAKE_PROFIT</code>
"Take profit limit orders are not supported for this symbol."	交易对上未启用 <code>TAKE_PROFIT_LIMIT</code>
"Price * QTY is zero or less."	<code>price</code> * <code>quantity</code> 太小
"IcebergQty exceeds QTY."	<code>icebergQty</code> 必须少于订单数量
"This action is disabled on this account."	联系客户支持；该帐户已禁用了某些操作。
"Unsupported order combination"	不允许组合 <code>orderType</code> , <code>timeInForce</code> , <code>stopPrice</code> , 和/或 <code>icebergQty</code> 。
"Order would trigger immediately."	与最后交易价格相比，订单的止损价无效。
"Cancel order is invalid. Check origClientId and orderId."	未发送 <code>origClientId</code> 或 <code>orderId</code> 。
"Order would immediately match and take."	<code>LIMIT_MAKER</code> 订单类型将立即匹配并进行交易，而不是纯粹的生成订单。
"The relationship of the prices for the orders is not correct."	<code>OCO</code> 订单中设置的价格不符合报价规则： The rules are: <code>SELL Orders</code> : Limit Price > Last Price > Stop Price <code>BUY Orders</code> : Limit Price < Last Price < Stop Price

Error message	Description
"OCO orders are not supported for this symbol"	OCO订单不支持该交易对
"Quote order qty market orders are not support for this symbol."	这个交易对，市价单不支持参数 quoteOrderQty
"Trailing stop orders are not supported for this symbol."	此symbol不支持 trailingDelta
"Order cancel-replace is not supported for this symbol."	此symbol不支持 POST /api/v3/order/cancelReplace

关于 POST /api/v3/order/cancelReplace 的错误

-2021 ORDER CANCEL-REPLACE PARTIALLY FAILED

收到该错误码代表撤单或者下单失败。

-2022 ORDER CANCEL-REPLACE FAILED.

收到该错误码代表撤单和下单都失败。

备注说明

请求参数

EMAIL地址

Email地址作为请求参数，需要转译(encode)。比如 alice@test.com 转换成 alice%40test.com