

REST API 参考文档

目录

1. 目录
2. 简介
3. 请求认证
 1. API Key 组成部分
 2. 请求接口的通用必要参数
 3. 签名算法
4. API Reference
 1. 普通 API【只读】
 2. 交易 API
5. 通用错误代码

简介

REST API 是为了方便用户进行自动化操作而提供的 HTTP API 接口，采用 RESTful 风格进行提供。

请求认证

用户可以在个人中心申请 API Key，具体申请方法参见个人中心页面。

用户申请到的 API Key 将直接与申请人绑定，也就是说，使用 API Key 进行请求访问时，将可以获得申请人账号的操作权限，所以，请谨慎保管您申请到的 API Key 签名密钥。

API Key 组成部分

一个完整的 API Key 包含一组 access key（访问 Key）和 secret key（签名 Key）。

- access key 为访问 Key，类似于“公钥”，用于标识用户的身份，access key 在申请的时候将会和唯一的一个用户进行绑定。
- secret key 为签名 Key，类似于“私钥”，用户必须妥善保管这个 Key，这个 Key 只会在用户申请 API Key 的时候向用户展示一次，后续无法再通过其他方式获得此 Key，如果遗失了这个 Key，请删除 API Key 后重新申请。

△ 请注意，secret key 相当于您的密码，获得这个 Key 后，配合对应的 access key 将可以获得 REST API 的完全访问权限，因此请一定妥善保管此 Key，以防止信息泄漏，甚至是资产损失！

请求接口的通用必要参数

所有的 API 接口均需要提供必要的 Query 参数（获取当前系统时间戳接口除外），这些参数保证请求是来自 API Key 持有者合法请求的。

如果在请求时，没有提供必要的参数，或是提供的参数不完整，请求将会被拒绝。

Query 参数列表

参数名	必填	参数含义	取值示例	说明
_	☑	时间戳	1532675556455	请求时刻的时间戳，精确到毫秒
_t		超时阈值	5000	默认值 5000，限制请求超时阈值，单位为毫秒，建议不超过 5 秒
access	☑	访问 Key	7gjqEQTKMvX80FbttztEW	申请的 API Key 中的 access key
sign	☑	签名	431f8f2c0b2017988a1b85bdf54bc95443dc02d7afdfcba460659b832ad841	使用签名算法计算获得的签名字符串

签名算法

由于互联网上传送数据并不安全，数据有可能被截获、篡改，为了安全，所有 API 接口都将对请求进行安全校验，确保请求来自 API Key 拥有者。

在请求接口的通用必要参数中，介绍了请求 API 所必须提供的参数，其中：

- _ 与 _t 用于确保此次请求仅在规定时间内有效，防止请求信息被拦截，在未来再次用来请求。
- sign 用于对请求的信息进行签名校验，此校验信息是通过用户的 secret key 生成的，不同的内容生成的 sign 将会完全不一样，这样可以防止请求的内容被篡改。

REST API 采用 Hmac-SHA256 算法对请求的数据进行签名校验，由于不同的内容（哪怕只是差了一个字母或是一个空格），得到的签名都会完全不一样，所以签名步骤必须正确，才能保证请求成功。

计算签名的详细步骤及示例

1. 首先，需要明确请求的接口，以及请求的参数。

这里，我们使用 `/api/x/v1/market/depth` 作为示例，请求 ETH/BTC 市场的深度数据。

所以，我们要请求的接口为 `/api/x/v1/market/depth`，Query 参数为 `symbol=ETH%2FBTC`。

参数	symbol
值	ETH%2FBTC

如果请求为 POST/PUT/PATCH，会在请求时带上 `body`，此时必须提供 `Content-Type: application/json` 或 `Content-Type: application/x-www-form-urlencoded` 请求头，并采用对应标准编码提交数据，`body` 的参数不参与签名。

不支持 `multipart/form-data` !

注：参数值请采用 [RFC3986](#) 标准进行 URL Encode，即空格使用 `%20`。

2. 添加通用的必要请求参数。

时间戳假设为 `1532681868919`，访问 Key 假设为 `7gjqEQQTKMvX80FbttztEW`。

参数	symbol	_	access
值	ETH%2FBTC	1532681868919	7gjqEQQTKMvX80FbttztEW

3. 接下来，需要对得到的参数进行签名，不过，在签名之前，还需要再追加两个参数，这两个参数仅参与签名运算，不加入请求参数中。

第一个参数为请求方法 `method`：`GET`，要求全部字母大写。第二个参数为请求的接口 `path`：`/api/x/v1/market/depth`。

注意，接口参数包含 `params` 参数，也就是最终请求的 `pathname`。

参数	symbol	_	access	method	path
值	ETH%2FBTC	1532681868919	7gjqEQQTKMvX80FbttztEW	GET	%2Fapi%2Fx%2Fv1%2Fmarket%2Fdepth

4. 至此，需要进行签名的所有参数都收集齐了，但由于不同内容得到但签名完全不同，为了得到相同的签名，我们需要先对参数按照规则进行排序。

我们使用字母升序的顺序对参数进行排序：`_ < access < method < path < symbol`。

5. 将排好序的参数列表按照 Query String 的形式进行拼接。

得到：`_=1532681868919&access=7gjqEQQTKMvX80FbttztEW&method=GET&path=%2Fapi%2Fx%2Fv1%2Fmarket%2Fdepth&symbol=ETH%2FBTC`。

6. 使用 Hmac-SHA256 算法和 `secret key` 签名 Key 对得到的字符串计算摘要签名。

假设 `secret key` 为 `3FFR01JhymbSCpVfCfAdjC`，则 Hmac-SHA256 得到的结果为：

`ab5ff7677f7927d2105bdcf4009231bb4b1ca5de2165d123fb21a4acd229ae0d`

7. 将参数放入请求。

所以我们最终要提交的 Query 参数为（此时的顺序不重要了）：

参 数	symbol	_	access	sign
值	ETH%2FBTC	1532681868919	7gjqEQQTKMvX80FbttztEW	ab5ff7677f7927d2105bdcf4009231bb4b1ca5de2165d123fb21a4acd229ae0d

所以最终请求的地址为：

`/api/x/v1/market/depth?`
`symbol=ETH%2FBTC&_=1532681868919&access=7gjqEQQTKMvX80FbttztEW&sign=ab5ff7677f7927d2105bdcf4009231bb4b1ca5de2165d123fb21a4acd229ae0d`

API Reference

普通 API 【只读】

GET `/api/x/v1/account/balance/{currency}` 获取指定币种余额

请求参数：

参数	类型	必填	示例	说明
----	----	----	----	----

参数	类型	必填	示例	说明
currency	params	<input checked="" type="checkbox"/>	BTC	币种

响应码:

- 200 OK

响应示例:

```
{
  "currency": "PTB", // 币种
  "balance": "1",    // 总计
  "available": "1",  // 可用
  "freezed": "0",    // 冻结
}
```

GET /api/x/v1/account/balance 获取所有币种余额

请求参数:

无

响应码:

- 200 OK

响应示例:

```
[
  {
    "currency": "PTB", // 币种
    "balance": "1",    // 总计
    "available": "1",  // 可用
    "freezed": "0",    // 冻结
  },
  {
    "currency": "BTC",
    "balance": "1",
    "available": "1",
    "freezed": "0",
  }
]
```

GET /api/x/v1/common/currencies 获取币种列表

请求参数:

无

响应码:

- 200 OK

响应示例:

```
[
  "BTC",
  "ETH"
]
```

GET /api/x/v1/common/symbols 获取交易对列表

请求参数:

无

响应码:

- 200 OK

响应示例:

```
[
  {
    "symbol": "ETH/BTC",      // 交易对
    "baseCurrency": "ETH",    // 基础币种
    "quoteCurrency": "BTC",   // 计价币种
    "priceDecimal": 8,        // 交易对价格精度
    "amountDecimal": 8,       // 交易对数量精度
    "fee": {                  // 手续费
      "buy": 0.0025,          // 买方手续费
      "sell": 0.0025,         // 卖方手续费
    },
    "priceRange": {          // 价格区间
      "buy": [1, 100],        // 买单允许价格区间
      "sell": [1, 100],       // 卖单允许价格区间
    },
  }
]
```

GET /api/x/v1/common/timestamp 获取当前系统时间戳

请求参数:

无

注: 此接口可以不包含任何授权校验参数直接请求

响应码:

- 200 OK

响应示例:

```
{
  "unix": 1532675557, // 精确到秒级别
  "ms": 1532675556541 // 精确到毫秒级别
}
```

GET /api/x/v1/market/depth 获取指定市场深度数据

请求参数:

参数	类型	必填	示例	说明
symbol	query	<input checked="" type="checkbox"/>	ETH/BTC	指定市场

响应码:

- 200 OK
- 404 Not Found

响应示例:

```
{
  "buyBills": [
    ["7.9784250", "819.83"],
    ["4.260962", "379.65"],
    ["4.112056", "37.58"]
  ],
  "askBills": []
}
```

GET /api/x/v1/market/kline 获取指定市场 K 线数据

请求参数:

参数	类型	必填	示例	说明
symbol	query	<input checked="" type="checkbox"/>	ETH/BTC	指定市场

参数	类型	必填	示例	说明
period	query	<input checked="" type="checkbox"/>	1	指定时间周期，可选： 1 , 5 , 15 , 30 , 60 , 120 , 240 , 360 , 720 , 1D , 3D , 1W
from	query		1527409940	起始时间（精确到秒），默认为当前时间
direction	query		older	获取方向，可选： older , newer ，默认为 older
max	query		300	最大接收数据条数，取值范围： [10, 1000] ，默认为 300

响应码：

- 200 OK
- 204 No Content
- 404 Not Found

响应示例：

```
[
  // ["开盘价", "最高价", "最低价", "收盘价", "成交量", "时间戳"]
  ["20", "20", "10", "10", "10", "1540259460"],
  ["20", "20", "20", "20", "20", "1540192740"],
  ["20", "20", "20", "20", "20", "1540191660"],
  ["20", "20", "20", "20", "20", "1540176780"],
  ["21", "21", "20", "20", "20", "1540176660"],
  ["21", "21", "21", "21", "21", "1539935520"],
  ["20", "21", "20", "21", "21", "1539929400"],
  ["11.672079", "20", "11.672079", "20", "60", "1539860640"],
  ["11.857871", "11.857871", "11.672079", "11.672079", "11.672079", "1539860280"],
  ["12.857871", "12.857871", "11.857871", "11.857871", "60.70417005", "1539860100"],
  ["12.857871", "12.857871", "12.857871", "12.857871", "25.715742", "1539860040"],
  ["14.369438", "14.369438", "12.857871", "12.857871", "65.1134114", "1539859260"],
  ["21", "21", "14.369438", "14.369438", "14.369438", "1539859140"],
  ["20", "21", "20", "21", "42", "1539858480"],
  ["18", "20", "18", "20", "40", "1539858240"],
  ["0", "18", "0", "18", "36", "1539857940"]
]
```

GET /api/x/v1/market/tickers 获取指定市场行情数据

请求参数：

参数	类型	必填	示例	说明
symbol	query	<input checked="" type="checkbox"/>	symbol=BTC%2FETH&symbol=BCH%2FETH	币种列表，多个币种按照 RFC6570 标准以重复提供参数的形式提供

注：签名的时候，签名列表顺序必须与提交时的顺序一致

响应码：

- 200 OK
- 404 Not Found

响应示例：

```
[
  {
    "change": "0.9607785570547472", // 24 小时内涨跌
    "timestamps": "1540329000", // 计时起点
    "open": "217309700", // 24 小时的开盘价
    "vol": "55662557130624800", // 24 小时的成交量（单位为基础币种，如 BTC/USDT 则单位为 BTC）
    "high": "799992900", // 24 小时最高价
    "low": "215609400", // 24 小时最低价
    "last": "426096200", // 最近一笔成交价
    "buy": "0", // 买一价格
    "sell": "0" // 卖一价格
  },
  {
    "change": "0.0",
    "timestamps": "1540329000",
    "open": "15521700",
    "vol": "99815256",
  }
]
```

```
    "high": "15521700",
    "low": "15521700",
    "last": "15521700",
    "buy": "0",
    "sell": "0"
  }
]
```

GET /api/x/v1/market/trade 获取指定市场成交记录

请求参数:

参数	类型	必填	示例	说明
symbol	query	<input checked="" type="checkbox"/>	ETH/BTC	指定市场
max	query		50	最大接收数据条数, 取值范围: [10, 300], 默认为 50

响应码:

- 200 OK
- 404 Not Found

响应示例:

```
// ["成交量", "成交价格", "买家促成或卖家促成", "时间戳", "id"]
[
  "2,18,ask,1539857940,1",
  "2,20,bid,1539858290,2",
  "2,21,bid,1539858535,3",
  "1,14.369438,ask,1539859155,4",
  "0.511,14.369438,ask,1539859309,5",
  "2.392,12.879462,ask,1539859309,6",
  "2.097,12.857871,ask,1539859309,7",
  "0.456,12.857871,ask,1539860096,8",
  "1.544,12.857871,ask,1539860096,9",
  "2.001,12.564925,ask,1539860157,10",
  "2.999,11.857871,ask,1539860157,11",
  "1,11.672079,ask,1539860336,12",
  "3,20,bid,1539860653,13",
  "1,21,ask,1539929441,14",
  "1,21,ask,1539935553,15",
  "1,20,ask,1540176705,16",
  "1,20,ask,1540176832,17",
  "1,20,ask,1540191714,18",
  "1,20,ask,1540192785,19",
  "1,10,ask,1540259507,20"
]
```

GET /api/x/v1/order/order 获取订单

请求参数:

参数	类型	必填	示例	说明
state	query	<input checked="" type="checkbox"/>	state=wait&state=done	类型, 可选: wait: 等待成交 / 未完全成交, done: 完全成交, cancel: 已撤销, canceling: 撤销进行中。 多个类型按照 RFC6570 标准以重复提供参数的形式提供
symbol	query		ETH/BTC	交易市场
type	query		buy	类型, 可选: buy, sell
from	query		2018-07-27T11:12:46.928Z	结束日期, ISO 8601 格式
to	query		2018-07-27T11:12:46.928Z	结束日期, ISO 8601 格式
page	query		1	页码, 取值范围: [1, +∞), 默认为 1

参数	类型	必填	示例	说明
limit	query		10	单页长度，取值范围：[10, 50]，默认为 10

响应码：

- 200 OK

响应示例：

```
{
  "list": [
    {
      "orderid": "5b23bd14b9d6ac00070a9a19", // 委托编号
      "trade_type": "sell", // 委托类型【buy / sell】
      "market": "EOS/BTC", // 交易对
      "memberid": 200003, // 用户 ID
      "price": "3", // 价格
      "volume": "44", // 未完成数量
      "origin_volume": "44", // 委托数量
      "done_at": null, // 委托完成时间
      "created_at": 1529068820, // 委托创建时间
      "updated_at": null, // 委托更新时间
      "source": "WEB", // 创建委托的来源【WEB / X-API】
      "ord_type": "limit", // 委托为限价类型
      "frezed": "44", // 当前资产锁定值，买单为 volume，买单为 price * volume
      "state": "wait", // 当前订单状态【wait, done, cancel, canceling】
      "origin_frezed": "44", // 委托的原始资产锁定值
      "funds_received": "0", // 收益 Σ(price(i) * volume(i))
      "trades_count": "0", // 交易次数
      "origin_amounts": "132",
      "amounts": "0",
      "already_volume": "0",
      "avg_price": "0"
    }
  ],
  "count": 1, // 数据总数
  "time": "2018-07-27T14:59:30.307Z" // 数据生成时间（当前时间）
}
```

GET /api/x/v1/order/order/{id} 获取单个订单信息

请求参数：

参数	类型	必填	示例	说明
id	params	☑	5aea22a41b8f35315ddd1069	订单 ID
detail	query		true	是否返回成交明细，默认为 false
page	query		1	页码，取值范围：[1, +∞)，当且仅当 detail 被设置为 true 时有意义，默认为 1
limit	query		10	单页长度，取值范围：[10, 50]，当且仅当 detail 被设置为 true 时有意义，默认为 10

响应码：

- 200 OK

响应示例：

```
{
  "orderid": "5b23bd14b9d6ac00070a9a19", // 委托编号
  "trade_type": "sell", // 委托类型【buy / sell】
  "market": "EOS/BTC", // 交易对
  "memberid": 200003, // 用户 ID
  "price": "3", // 价格
  "volume": "44", // 未完成数量
  "origin_volume": "44", // 委托数量
  "done_at": null, // 委托完成时间
  "created_at": 1529068820, // 委托创建时间
  "updated_at": null, // 委托更新时间
  "source": "WEB", // 创建委托的来源【WEB / X-API】
}
```

```
"ord_type": "limit", // 委托为限价类型
"freezed": "44", // 当前资产锁定值, 卖单为 volume, 买单为 price * volume
"state": "wait", // 当前订单状态【wait, done, cancel, canceling】
"origin_freezed": "44", // 委托的原始资产锁定值
"funds_received": "0", // 收益 Σ(price(i) * volume(i))
"trades_count": "0", // 交易次数
"origin_amounts": "132",
"amounts": "0",
"already_volume": "0",
"avg_price": "0",
"detail": { // 订单成交明细 (仅当请求参数 `detail` 为 `true` 时返回此字段)
  "data": [
    {
      "price": "3", // 成交价
      "tradeid": "5b1fba5611db110007f310e05b23bd14b9d6ac00070a9a19", // 交易 ID
      "volume": "44", // 成交量
      "trade_price": "132", // 成交额
      "sellerid": "200003", // 买单用户
      "buyerid": "200106", // 卖单用户
      "ask_id": "5b23bd14b9d6ac00070a9a19", // 卖委托 ID
      "bid_id": "5b1fba5611db110007f310e0", // 买委托 ID
      "trend": 0, // 成交趋势, 相对于前一次交易【0: 价格相同, 1: 价格上涨, 2: 价格下跌】
      "type": 1, // 【保留参数】
      "trade_type": "sell", // 促成交易类型
      "market": "EOS/BTC", // 交易对
      "created_at": 1529068821, // 交易创建时间戳
      "updated_at": 1529068821, // 交易更新时间戳
      "bid_fees": "0", // 买单使用当前币种结算的手续费
      "bid_ptb": "0.03583883", // 买单使用平台币结算的手续费
      "bid_rate": 0.0025, // 买单使用当前币种结算的手续费率
      "bid_ptb_rate": 0.00125, // 买单平台币结算手续费费率
      "ask_fees": "0", // 卖单使用当前币种结算的手续费
      "ask_ptb": "0.03583883", // 卖单使用平台币结算的手续费
      "ask_rate": 0.0025, // 卖单使用当前币种结算的手续费率
      "ask_ptb_rate": 0.00125, // 卖单使用平台币结算的手续费
      "ord_type": "sell",
      "order_type": "sell"
    }
  ],
  "total": 1
}
}
```

GET /api/x/v1/order/order/trade 查询历史成交记录

请求参数:

参数	类型	必填	示例	说明
symbol	query		ETH/BTC	交易市场
type	query		buy	类型, 可选: buy, sell
from	query		2018-07-27T11:12:46.928Z	结束日期, ISO 8601 格式
to	query		2018-07-27T11:12:46.928Z	结束日期, ISO 8601 格式
page	query		1	页码, 取值范围: [1, +∞), 默认为 1
limit	query		10	单页长度, 取值范围: [10, 50], 默认为 10

响应码:

- 200 OK

响应示例:

```
[
  {
    "_id": "5b52d0442695d2f285a230eb",
    "createdAt": 1532153923,
    "amount": 61.83,
    "fees": "0.36580438 BTC",
    "id": "5b51c8232011e0000798739b5b52d043cb11870007322ee2",
  }
]
```



```
[
  {
    "orderId": "5b51c8232011e0000798739b",
    "price": 44,
    "symbol": "EOS/BTC",
    "type": "buy"
  },
  {
    "_id": "5b51c83a2695d2f28530fb82",
    "createdAt": 1532086330,
    "amount": 5,
    "fees": "0.09607503 EOS",
    "id": "5b51c8232011e0000798739b5b51c839975b3f00072998e4",
    "orderId": "5b51c839975b3f00072998e4",
    "price": 44,
    "symbol": "EOS/BTC",
    "type": "sell"
  }
]
```

交易 API

POST /api/x/v1/order/order 下单

请求参数:

参数	类型	必填	示例	说明
symbol	body	☑	ETH/BTC	交易市场
type	body	☑	buy	成交方向, 可选: buy, sell
price	body	☑	1.234	价格
amount	body	☑	1.234	数量

响应码:

- 200 OK
- 400 Bad Request
- 401 Forbidden
- 404 Not Found

响应示例:

```
{
  "orderid": "5b5b3a5fa93fef000655e678"
}
```

DELETE /api/x/v1/order/order/{id} 撤单

请求参数:

参数	类型	必填	示例	说明
id	params	☑	5aea22a41b8f35315ddd1069	订单 ID
symbol	query	☑	ETH/BTC	交易市场

响应码:

- 201 Created (Deleted)
- 400 Bad Request
- 401 Forbidden
- 404 Not Found

响应示例:

正常响应没有 body

通用错误代码

- ✖ 400 Bad Request

通常由于请求时提供的参数格式错误、缺少或存在多余参数，您可以根据响应 Body 中给出的提示以检查请求参数

- ✕ 401 Unauthorized

通常由于客户端提供的访问 key（access）或签名（sign）不正确，或者是 API 权限不足

- ✕ 403 Forbidden

通常由于用户账号被后台封禁

- ✕ 404 Not Found

详情根据 message 而定

- ✕ 408 Timed Out

通常由于请求中指定的时间戳超过有效阈值，或者时间戳格式错误

- ✕ 418 I'm a Teapot

我现在不能煮咖啡，请稍后再试

此状态用于实现 RFC 2324 咖啡壶控制协议，但是由于某些未知原因，咖啡壶尚不能工作

通常由于咖啡壶的主人只允许受邀请的 VIP 客人使用他的咖啡壶，而其他客人必须购买自己的咖啡壶

- ✕ 428 Precondition Required

通常由于请求中没有包含时间戳、访问/签名 Key

- ✕ 429 Too Many Requests

通常由于短期内多次请求，达到了后台配置的单位时间内请求次数限制阈值