**Cybex API Server用户手册**

目录

[1 概述 3](#_Toc536547552)

[2 拓扑图 3](#_Toc536547553)

[3 Cyb-signer 4](#_Toc536547554)

[3.1 使用Linux系统 4](#_Toc536547555)

[3.1.1 预装软件 4](#_Toc536547556)

[3.1.2 下载cyb-signer源代码并编译 4](#_Toc536547557)

[3.1.3 配置cyb-signer 5](#_Toc536547558)

[3.1.4 获取执行文件 5](#_Toc536547559)

[3.1.5 启动和暂停 cyb-signer 6](#_Toc536547560)

[3.1.6 签名操作 6](#_Toc536547561)

[3.2 使用Windows系统 8](#_Toc536547562)

[3.3 使用Mac系统 9](#_Toc536547563)

[4 API Server 9](#_Toc536547564)

[5 常见问题 17](#_Toc536547565)

[5.1 如何查看Cybex 账户名（account name）及账户ID（account id） 17](#_Toc536547566)

[5.2 如何查看您的私钥（private key） 18](#_Toc536547567)

[5.3 什么是交易手续费 18](#_Toc536547568)

[5.4 更多相关信息 19](#_Toc536547569)

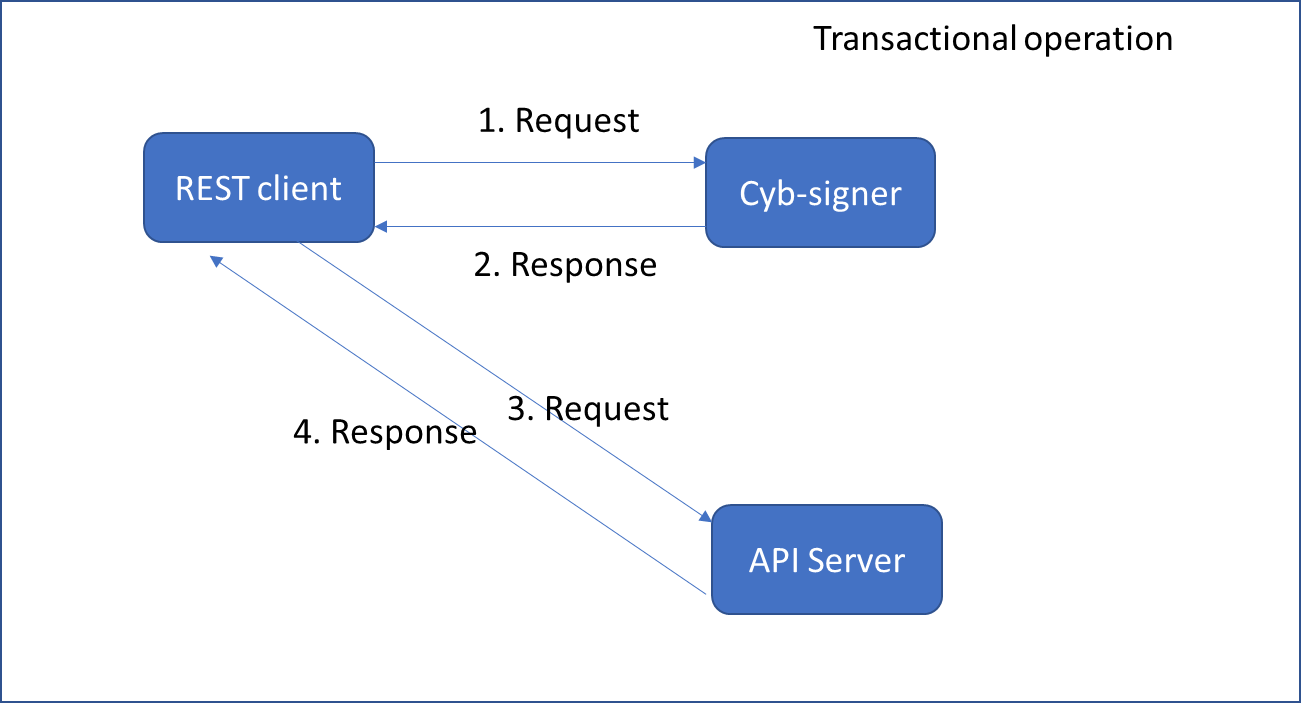
[5.5 如何安装JDK或JRE 19](#_Toc536547570)

# 概述

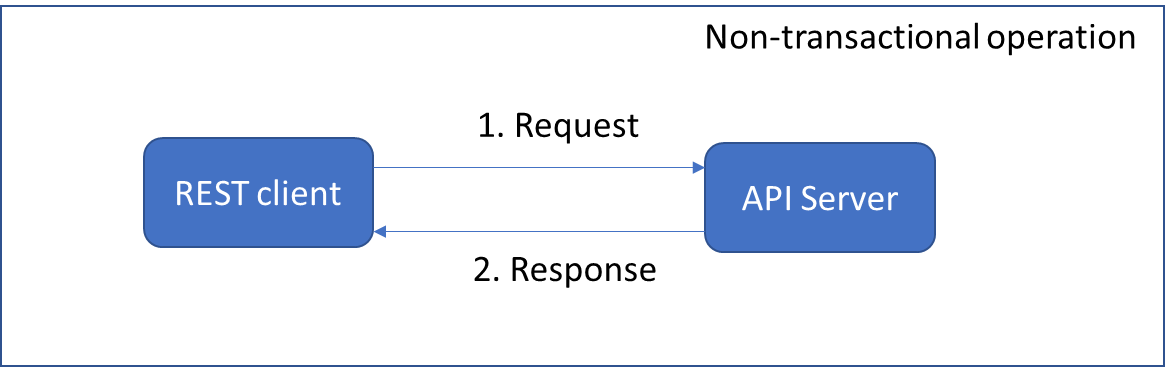
本文旨在帮助用户完成本地cyb-signer搭建并对如何使用cybex API Server提供指导。

# 拓扑图

交易操作（Transactional operation）在发送至API Server前，需要使用用户的私钥进行签名。用户可以使用cyb-signer返回的结果直接转发至API Server。



对于非交易操作（Non-transactional operation），可以直接通过 REST方式与API Server通讯。

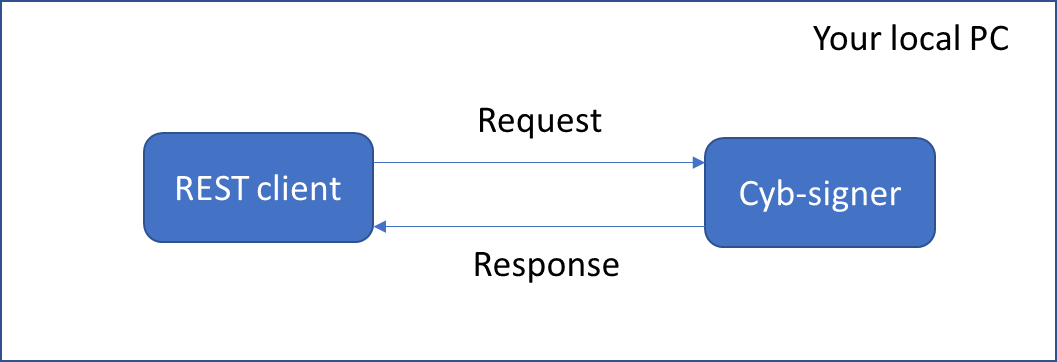


当前支持的所有操作如下表所示：

|  |  |
| --- | --- |
| **操作类型** | **操作名称** |
| 交易性 | 创建委单 |
| 取消委单 |
| 取消所有委单 |
| 非交易性 | 查询委单状态 |
| 查询交易状态 |
| 查询交易所相关信息 |
| 查询用户仓位 |
| 查询交易对的委单队列 |
| 查询交易对的K线图信息 |

# Cyb-signer

本章将具体介绍搭建本地cyb-signer的操作步骤。该程序将在用户本地环境内创建一个包含用户私钥的REST Server，以确保在不会暴露用户私钥的情况下，系统可以通过相对简单的方式实现交易操作的签名过程。用户可以使用任何REST客户端进行签名。



## 使用Linux系统

用户可以参照以下步骤搭建在Linux环境内的cyb-signer，所有操作界面截图均来自Ubuntu 16.04操作系统。

### 预装软件

在开始搭建环境前，请确保您已安装以下软件：

|  |  |  |
| --- | --- | --- |
| 目的 | 软件 | 版本 |
| 直接使用cyb-signer | JRE | 1.8 或更高版本 |
| Git | 2.7 或更高版本 |
| 进行源代码编译及启动cyb-signer | JDK | 1.8或更高版本 |
| Git | 2.7或更高版本 |
| Maven | 3.0或更高版本 |

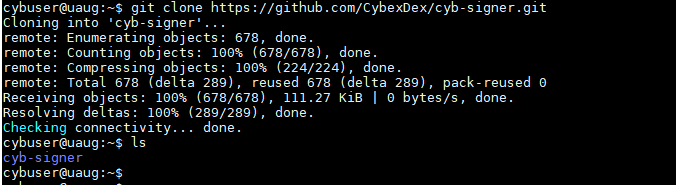
JRD及JDK的具体安装步骤请参考5.5 章节。

### 下载cyb-signer源代码并编译

您可以从GitHub的主仓库和镜像仓库中获取所有代码内容：

|  |  |  |
| --- | --- | --- |
| 资源类型 | Url | 备注 |
| 主仓库 | 源代码: https://github.com/CybexDex/cyb-signer.git |  |
| 版本发布: https://github.com/CybexDex/cyb-signer/releases |  |
| 镜像仓库 | 源代码: https://git.coding.net/david\_lu/cyb-signer.git | 在中国内地使用，速度较快 |
| 版本发布: https://coding.net/u/david\_lu/p/cyb-signer/git/releases |

请通过 “git clone” 命令将代码下载到您本地。



### 配置cyb-signer

请在阁下电脑的cyb-signer/scripts文件夹下的env.properties文件内配置参数。

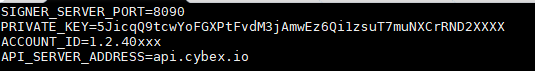
以下为部分参数示例：

SIGNER\_SERVER\_PORT=8090 （本机未用的端口）

PRIVATE\_KEY=5JicqQ9tcwYoFGXPtFvdM3jAmwEz6Qi1zsuT7muNXCrRND2XXXX （您的私钥）

ACCOUNT\_ID=1.2.xxxxx （您的cybex账户ID）

API\_SERVER\_ADDRESS=api.cybex.io （固定值）



注意：

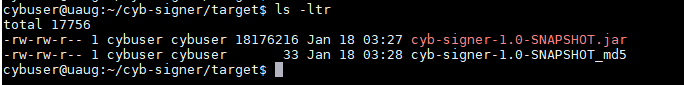
* ACCOUNT ID与ACCOUNT NAME不同，ACCOUNT ID的格式为1.2.xxxx。请参照5.1中的具体步骤获取该信息。
* 私钥（Private Key）不是您登录cybex网站时填写的密码，请参照5.2中的具体步骤获取该信息。

### 获取执行文件

您可以通过从版本发布URL中下载jar包或者从源代码中进行编译。

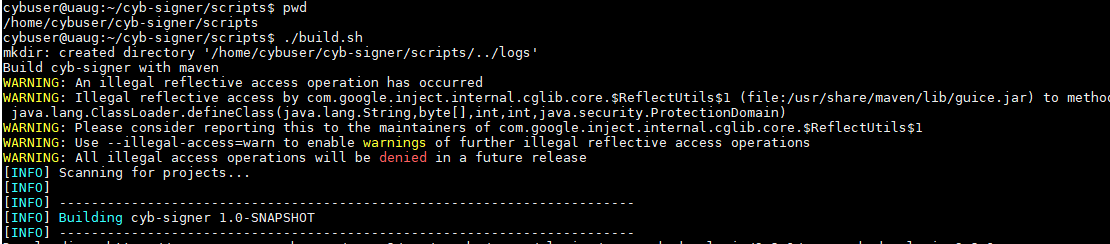
#### 从版本发布URL中下载

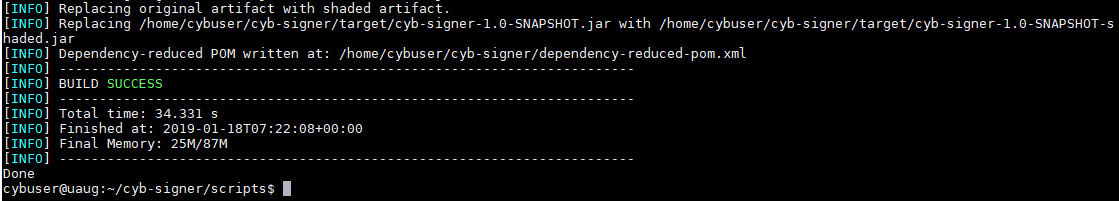
如果您希望直接使用jar包，可以从3.1.2中提供的URL下载jar包并将其放在“cyb-signer/target”目录内。



#### 从源代码进行编译

如果您希望通过源代码编译jar包，你可以在“cyb-signer/scripts” 目录下执行“build.sh”命令。由于maven需要下载一定数量的文件，初次编译可能会花费较长的时间。





### 启动和暂停 cyb-signer

使用start.sh/stop.sh命令可以启动或暂停cyb-signer。一旦cyb-signer启动，除了需要修改参数值或者更新cyb-signer源码外，您无需手动进行重启。





为了增加签名过程的安全性，cyb-signer和您的REST客户端必须部署在同一台机器内。cyb-signer将会拒绝所有没有使用localhost为域名的http请求，例如，<http://192.168.1.10:8090/signer/v1/newOrder> 请求将不会成功。

### 签名操作

现在您可以通过REST客户端向cyb-signer发送签名请求了。

* 创建委单

POST <http://localhost:8090/signer/v1/newOrder>

|  |  |
| --- | --- |
| 参数 | 示例值 |
| assetPair | ETH/USDT, EOS/USDT |
| price | 146.67 |
| quantity | 10 |
| side | buy or sell |

|  |  |
| --- | --- |
| 调用示例 |  |
| 请求 | curl --data '{"assetPair": "ETH/USDT","price": 146.67,"quantity": 10,"side": "buy"}' http://localhost:8090/signer/v1/newOrder |
| 返回 | {  "transactionType" : "NewLimitOrder",  "transactionId" : "40c275d77233c5761472673bd0532150dde723a0",  "refBlockNum" : 18956,  "refBlockPrefix" : 507092400,  "txExpiration" : 1546919505,  "fee" : {  "assetId" : "1.3.0",  "amount" : 55  },  "seller" : "1.2.40658",  "amountToSell" : {  "assetId" : "1.3.27",  "amount" : 1466699999  },  "minToReceive" : {  "assetId" : "1.3.2",  "amount" : 10000000  },  "expiration" : 1546991999,  "signature" : "1c7a00b61039b6bc1ace93916446207a28a69e60a97fbd6b26  7518754a13e7adca463239452b0e00aa90b71b30a57ea4d65fa4cee44a30ab  2c7841fb5cd0145913",  "fill\_or\_kill" : 0  } |

* 取消委单

POST <http://localhost:8090/signer/v1/cancelOrder>

|  |  |
| --- | --- |
| 参数 | 示例值 |
| originalTransactionId | 88cecaa11b8584fb21243cd57ed2227e7c181452 |

|  |  |
| --- | --- |
| 调用示例 |  |
| 请求 | curl --data '{"originalTransactionId": "88cecaa11b8584fb21243cd57ed2227e7c181452"}' http://localhost:8090/signer/v1/cancelOrder |
| 返回 | {  "transactionType" : "Cancel",  "transactionId" : "e3a8b5c5afd2daa038707aa2e9e2d60701d726c0",  "originalTransactionId" : "88cecaa11b8584fb21243cd57ed2227e7c181452",  "refBlockNum" : 18956,  "refBlockPrefix" : 507092400,  "txExpiration" : 1546950595,  "orderId" : "0",  "fee" : {  "assetId" : "1.3.0",  "amount" : 5  },  "feePayingAccount" : "1.2.40658",  "signature" : "1c313f1f34c7975e3999d483ddf1d2d2322751593b82f3f087f  987e95a03b1639d696e2348105f2a9070c11c974b2c94ed616d955  59108d7714f516fa5ee55d559"  } |

* 取消所有委单

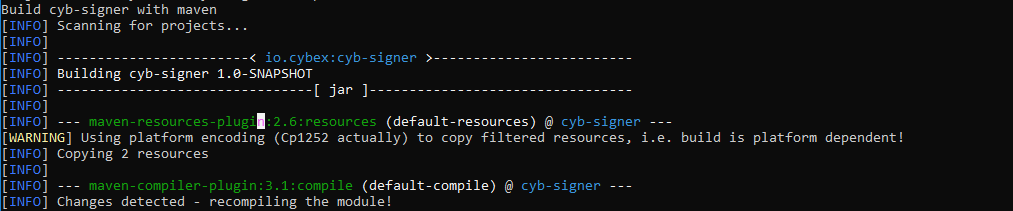
POST <http://localhost:8090/signer/v1/cancelAll>

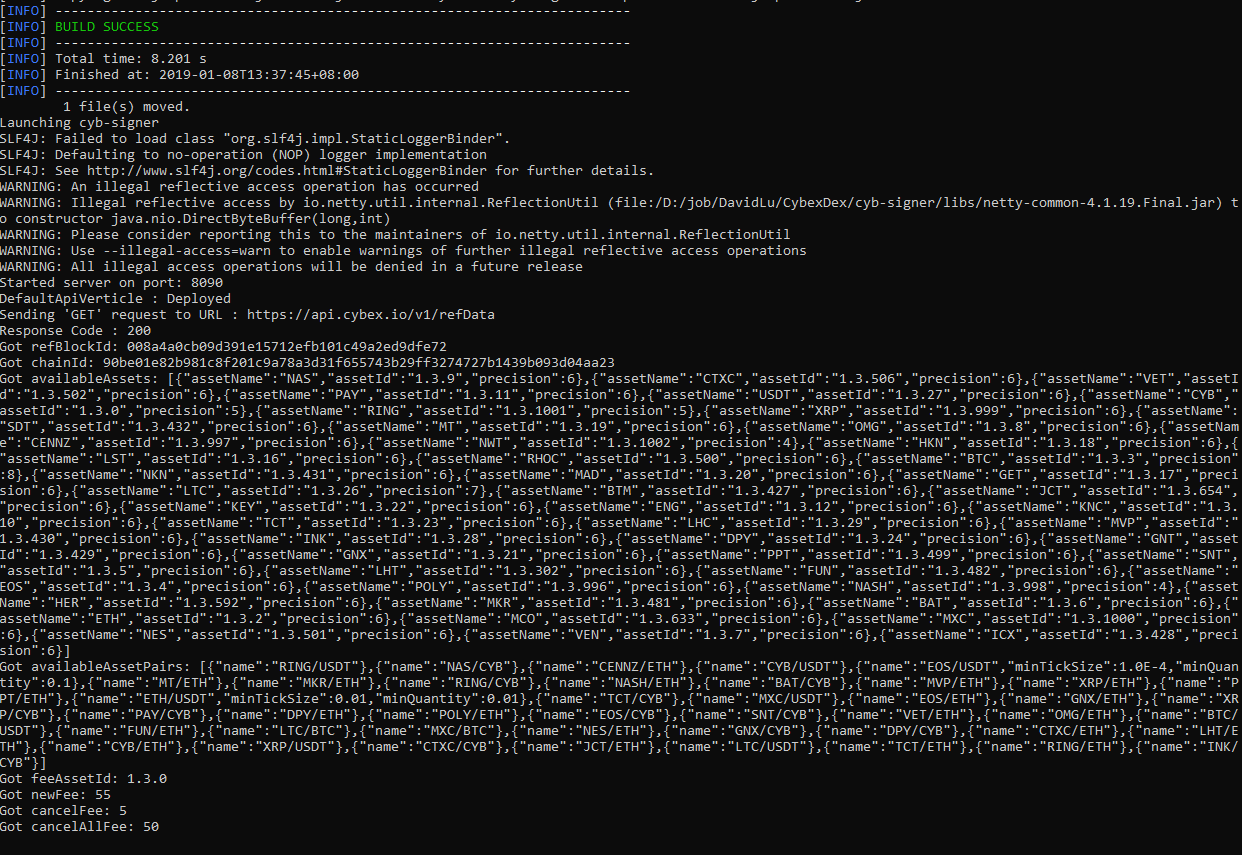
|  |  |
| --- | --- |
| 参数 | 示例值 |
| assetPair | ETH/USDT. 您可以使用“CYB/CYB”来取消您所有交易对的委单。 |

|  |  |
| --- | --- |
| 调用示例 |  |
| 请求 | curl --data '{"assetPair": "ETH/USDT"}' http://localhost:8090/signer/v1/cancelAll |
| 返回 | {  "transactionType" : "CancelAll",  "transactionId" : "9616b03391ccf0107c58326f2674c4b51a7bf284",  "refBlockNum" : 18956,  "refBlockPrefix" : 507092400,  "txExpiration" : 1547007167,  "fee" : {  "assetId" : "1.3.0",  "amount" : 50  },  "seller" : "1.2.40658",  "sellAssetId" : "1.3.2",  "recvAssetId" : "1.3.27",  "signature" : "1c3435fd231625f61a516e44bcb193f6344441974cf76def07  1ea01e82a1a2d0043e44ee28993d523fa13bad5445a297cdeef6f0b6a8650  7e8cbba7841ec6a9010"  } |

## 使用Windows系统

您可以参照3.1的相关步骤完成Windows系统内的搭建工作，当您完成参数配置后，可以通过“start.bat”启动程序。当您需要关闭程序时，仅需关闭窗口即可。



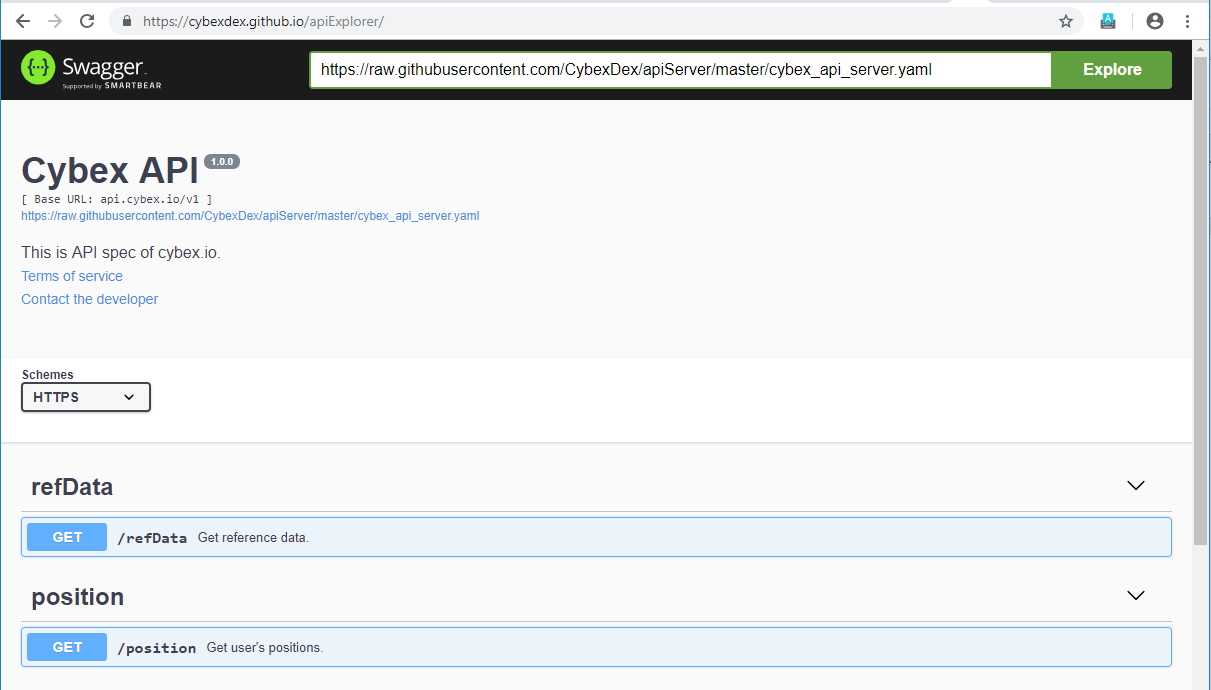


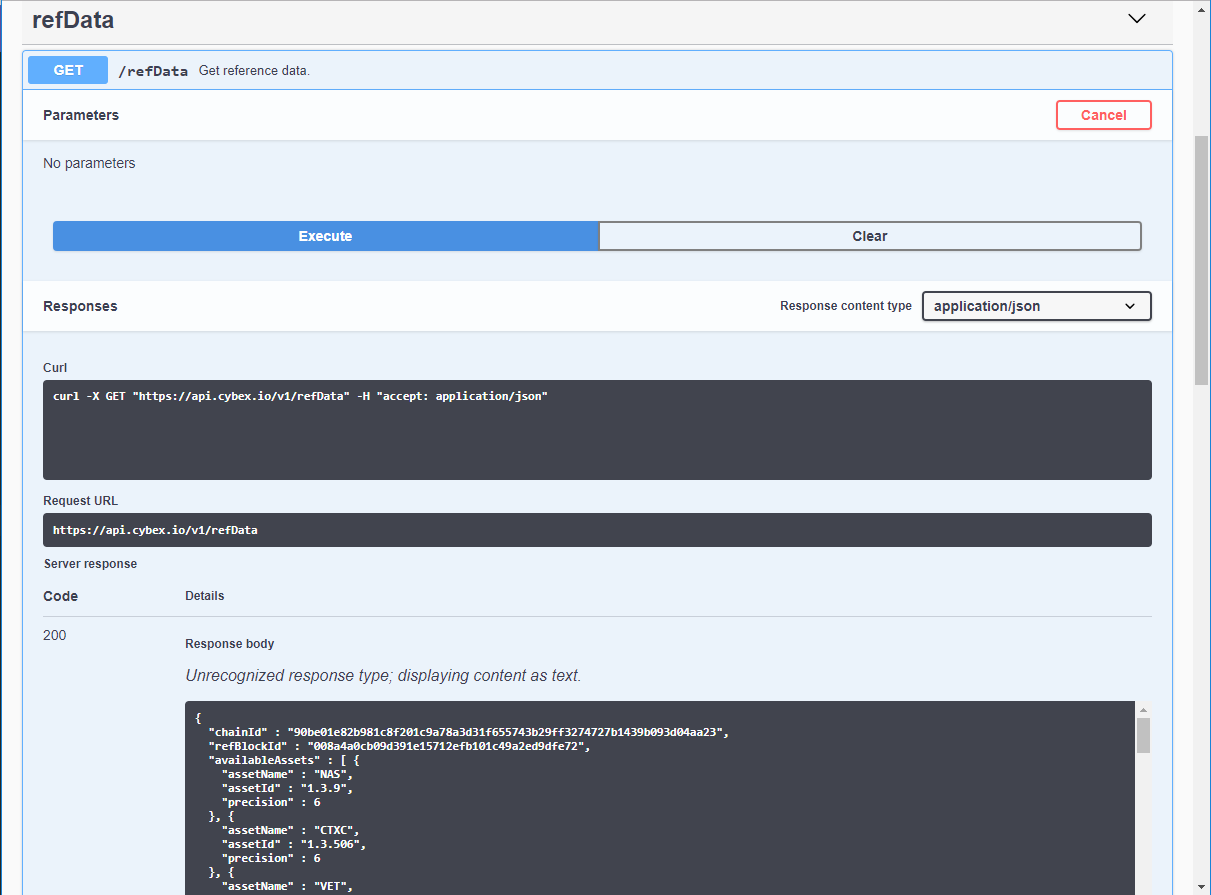
## 使用Mac系统

您如果您使用的是Mac系统，可以参照3.1中除了脚本相关步骤。一旦您在env.properties文件内完成了配置，就可以通过“build.sh”命令编译代码，并且通过“startMac.sh/stopMac.sh”命令来启动或暂停cybex-signer。

# API Server

根据第2章的相关描述，API Server会执行交易操作和进行相关的查询。您可以通过api浏览器查看各个API的参数详情以及执行操作： <https://cybexdex.github.io/apiExplorer>





接下来，将为您详细介绍每个API的参数并提供调用示例。

* 获取交易所相关信息

GET <https://api.cybex.io/v1/refData>

查询交易所相关的静态数据信息，例如：链ID、相关区块ID、可用资产类型、可用交易对、费用信息。

|  |  |
| --- | --- |
| 参数 | 示例值 |
| 无 |  |

|  |  |
| --- | --- |
| 调用示例 |  |
| 请求 | curl https://api.cybex.io/v1/refData |
| 返回 | {  "chainId" : "90be01e82b981c8f201c9a78a3d31f655743b29ff3274727b1439b093d04aa23",  "refBlockId" : "008a4a0cb09d391e15712efb101c49a2ed9dfe72",  "availableAssets" : [{  "assetName" : "ETH",  "assetId" : "1.3.2",  "precision" : 6  }, {  "assetName" : "USDT",  "assetId" : "1.3.27",  "precision" : 6  }, {  "assetName" : "CYB",  "assetId" : "1.3.0",  "precision" : 5  }, {  "assetName" : "EOS",  "assetId" : "1.3.4",  "precision" : 6  },  ......  ],  "availableAssetPairs" : [{  "name" : "EOS/USDT",  "minTickSize" : 0.0001,  "minQuantity" : 0.1  }, {  "name" : "ETH/USDT",  "minTickSize" : 0.01,  "minQuantity" : 0.01  }  ......  ],  "fees" : {  "feeAssetId" : "1.3.0",  "newFee" : 55,  "cancelFee" : 5,  "cancelAllFee" : 50  }  } |

* 执行交易

POST https://api.cybex.io/v1/transaction

用户需要将cyb-singer的返回作为参数调用该API。

|  |  |
| --- | --- |
| 参数 | 示例值 |
| Json格式的字符串 | cyb-signer的返回内容。交易包括：创建委单、取消委单和取消全部委单。 |

|  |  |
| --- | --- |
| 调用示例 |  |
| 请求 | curl --data '{ "transactionType": "NewLimitOrder", "transactionId": "040c6466dc0cebb8de40520ebb7346fe0e446b35", "refBlockNum": 18956, "refBlockPreix": 507092400, "txExpiration": 1546926850, "fee": { "assetId": "1.3.0", "amount": 55 }, "seller": "1.2.40658", "amountToSell": { "assetId": "1.3.27", "amount": 277000 }, "minToReceive": { "assetId": "1.3.4", "amount": 100000 }, "expiration": 1546991999, "signature": "1b477931ebb89b39f7e1f1953c9ead66f1c93eee8183f4334729e8  f19dceab2a5000af53728952dd73bc3636ab0663cc8f0293714d4c4  ed8f3399c5a50b9da2816", "fill\_or\_kill": 0}' https://api.cybex.io/v1/transaction |
| 正确返回 | {  "Status": "Successful",  "orderSequence": 529016,  "signature": "1b477931ebb89b39f7e1f1953c9ead66f1c93eee8183f4334729e8f1  9dceab2a5000af53728952dd73bc3636ab0663cc8f0293714d4c4ed8f3399c5a50b9  da2816",  "time": "2019-01-08T05:53:19.049071Z"  } |
| 错误返回 | {  "Status" : "Failed",  "Code" : 1005,  "Message" : "Duplicated transaction",  "signature" : "1b477931ebb89b39f7e1f1953c9ead66f1c93eee8183f4334729e8f19dceab2a  5000af53728952dd73bc3636ab0663cc8f0293714d4c4ed8f3399c5a50b9da28  16"  } |

* 查询委单状态

GET https://api.cybex.io/v1/order

通过该API可以查询指定cybex账户的委单状态。

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必填 | 示例值 |
| accountName | STRING | YES | 您的cybex账户名 |
| reverse | BOOL | NO | 当值为true时，表示按照委单创建时间逆序显示，即距离当前时间最近的记录显示在最前。默认值为true。 |
| assetPair | STRING | NO | 交易对，如：ETH/USDT |
| orderStatus | STRING | NO | 委单状态，如：FILLED。查询多个状态时，使用英文逗号“,”进行分隔，如：FILLED, OPEN |
| startTime | STRING | NO | 起始时间，格式为：YYYY-MM-DDTHH:mm:ss.ssssssZ，如：2018-01-07T01:20:48.647910Z |
| endTime | STRING | NO | 结束时间，格式为：YYYY-MM-DDTHH:mm:ss.ssssssZ，如：2018-01-07T01:20:48.647910Z |
| start | INT | NO | 显示的起始记录编号 |
| limit | INT | NO | 最大返回记录数 |
| transactionId | STRING | NO | 指定transaction ID |

|  |  |
| --- | --- |
| 调用示例 |  |
| 请求 | curl 'https://api.cybex.io/v1/order?accountName=XXXX&reverse=true&orderStatus=FILLED,CANCELED' |
| 返回 | [ {  "accountName" : "XXXX",  "orderSequence" : 529016,  "orderStatus" : "FILLED",  "assetPair" : "EOS/USDT",  "side" : "buy",  "price" : 2.77,  "quantity" : 0.1,  "filledQuantity" : 0.102437,  "averagePrice" : 2.7041010572,  "transactionId" : "040c6466dc0cebb8de40520ebb7346fe0e446b35",  "createTime" : "2019-01-08T05:53:19.046342Z",  "lastUpdateTime" : "2019-01-08T05:53:21.806097Z"  }, {  "accountName" : "XXXX",  "orderSequence" : 169409,  "orderStatus" : "FILLED",  "assetPair" : "EOS/USDT",  "side" : "buy",  "price" : 2.77,  "quantity" : 0.1,  "filledQuantity" : 0.100264,  "averagePrice" : 2.762706455,  "transactionId" : "e447bca7d48412d8fc4aecc91224d008a4946145",  "createTime" : "2019-01-03T01:33:39.501859Z",  "lastUpdateTime" : "2019-01-03T01:33:42.722457Z"  }] |

* 查询交易状态

GET https://api.cybex.io/v1/trade

通过该API可以查询指定cybex账户的交易状态。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 类型 | 是否必填 | 示例值 | |
| accountName | STRING | YES | | 您的cybex账户名 |
| reverse | BOOL | NO | | 但值为true时，表示按照交易上链时间逆序显示，即距离当前时间最近的记录显示在最前。默认值为true。 |
| assetPair | STRING | NO | | 交易对，如：ETH/USDT |
| startTime | STRING | NO | | 起始时间，格式为：YYYY-MM-DDTHH:mm:ss.ssssssZ，如：2018-01-07T01:20:48.647910Z |
| endTime | STRING | NO | | 结束时间，格式为：YYYY-MM-DDTHH:mm:ss.ssssssZ，如：2018-01-07T01:20:48.647910Z |
| start | INT | NO | | 显示的起始记录编号 |
| limit | INT | NO | | 最大返回记录数 |

|  |  |
| --- | --- |
| 调用示例 |  |
| 请求 | curl 'https://api.cybex.io/v1/trade?accountName=XXXXXX&reverse=true' |
| 返回 | [ {  "accountName" : "XXXX",  "orderSequence" : 529016,  "chainOrderId" : 395937330,  "assetPair" : "EOS/USDT",  "side" : "buy",  "orderPrice" : 2.77,  "orderQuantity" : 0.1,  "tradePrice" : 2.7041010572,  "tradeQuantity" : 0.102437,  "blockTime" : "2019-01-08T05:53:21.000000Z"  }, {  "accountName" : "XXXX",  "orderSequence" : 169409,  "chainOrderId" : 384202079,  "assetPair" : "EOS/USDT",  "side" : "buy",  "orderPrice" : 2.77,  "orderQuantity" : 0.1,  "tradePrice" : 2.762706455,  "tradeQuantity" : 0.100264,  "blockTime" : "2019-01-03T01:33:42.000000Z"  }, {  "accountName" : "XXXX",  "orderSequence" : 792863,  "chainOrderId" : 382273197,  "assetPair" : "ETH/USDT",  "side" : "sell",  "orderPrice" : 143.1,  "orderQuantity" : 0.01,  "tradePrice" : 143.1799,  "tradeQuantity" : 0.01,  "blockTime" : "2019-01-02T07:52:54.000000Z"  } ] |

* 查询用户仓位

GET https://api.cybex.io/v1/position

通过该API查询指定cybex账户的当前仓位。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 类型 | 是否必填 | 示例值 | |
| accountName | STRING | YES | | 您的cybex账户名 |

|  |  |
| --- | --- |
| 调用示例 |  |
| 请求 | curl 'https://api.cybex.io/v1/position?accountName=XXXX' |
| 返回 | {  "accountName" : "XXXX",  "positions" : [ {  "assetName" : "CYB",  "quantity" : 3290.30569  }, {  "assetName" : "ETH",  "quantity" : 0.377978  }, {  "assetName" : "USDT",  "quantity" : 4.376498  }, {  "assetName" : "EOS",  "quantity" : 0.202701  } ],  "time" : "2019-01-08T06:19:42.025291Z"  } |

* 查询指定交易对的委单队列

GET https://api.cybex.io/v1/orderBook

通过该API查询指定交易对的委单队列。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 类型 | 是否必填 | 示例值 | |
| assetPair | STRING | YES | | 如：ETH/USDT, EOS/USDT |
| limit | INT | YES | | 最大返回记录数 |

|  |  |
| --- | --- |
| 调用示例 |  |
| 请求 | curl 'https://api.cybex.io/v1/orderBook?assetPair=ETH/USDT&limit=3' |
| 返回 | {  "assetPair": "ETH/USDT",  "bids": [  [  "146.31", // Price  "5.0", // Quantity  "731.55" // Precise notional from price x quantity  ],  [  "146.3",  "3.192823",  "467.110114"  ],  [  "146.28",  "3.0",  "438.84"  ]  ],  "asks": [  [  "146.36",  "3.902031",  "571.062364"  ],  [  "146.39",  "3.19903",  "468.306001"  ],  [  "146.41",  "3.742901",  "547.960743"  ]  ],  "time": "2019-01-08T06:22:37.47311Z"  } |

* 查询K线图

GET https://api.cybex.io/v1/klines

通过该API查询指定交易对的K线图信息。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 类型 | 是否必填 | 示例值 | |
| assetPair | STRING | YES | | 如：ETH/USDT, EOS/USDT |
| interval | ENUM | YES | | 1m, 3m, , 5m, 15m, 30m, 1h, 2h, 4h, 6h, 8h, 12h, 1d, 3d, 1w, 1M |
| startTime | LONG | NO | | 起始时间，格式为：YYYY-MM-DDTHH:mm:ss.ssssssZ，如：2018-01-07T01:20:48.647910Z |
| endTime | LONG | NO | | 结束时间，格式为：YYYY-MM-DDTHH:mm:ss.ssssssZ，如：2018-01-07T01:20:48.647910Z |
| limit | INT | NO | | 最大返回记录数 |

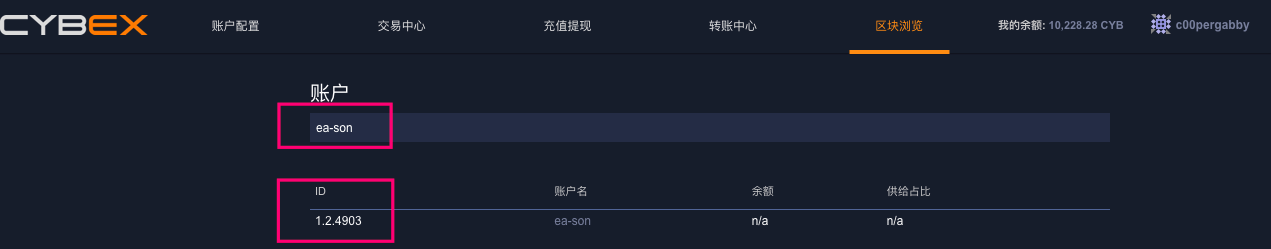
|  |  |
| --- | --- |
| 调用示例 |  |
| 请求 | curl 'https://api.cybex.io/v1/klines?assetPair=EOS/USDT&interval=1m&limit=2' |
| 返回 | [  [  1546928820000, // Open time  "2.70150000", // Open price  "2.70240000", // High  "2.70150000", // Low  "2.70200000", // Close  "282.91000000", //Volume  1546928879999, // Close time  "764.40811500", // Quote asset volume  3, // Number of trades  "64.45000000", // Taker buy base asset volume  "174.16968000", // Taker buy quote asset volume  "0" // Reserved  ],  [  1546928880000,  "2.70200000",  "2.70200000",  "2.70160000",  "2.70160000",  "163.48000000",  1546928939999,  "441.70876800",  2,  "163.48000000",  "441.70876800",  "0"  ]  ] |

# 常见问题

## 如何查看Cybex 账户名（account name）及账户ID（account id）

Account name 是您在注册Cybex时设置的用户名。您可以通过访问<https://olddex.cybex.io>并登录后，查看其对应的account id，具体步骤是：进入“区块浏览”，选择“账户”，输入您要查找的账户名，在列表内查看该账户的账户id。

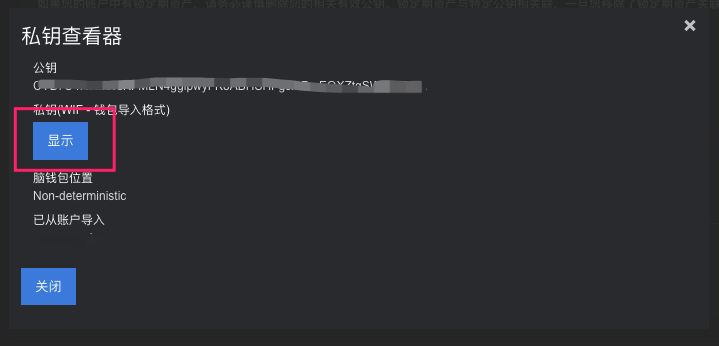




## 如何查看您的私钥（private key）

登录<https://olddex.cybex.io>，进入“账户配置”，选择“权限”。在“活跃权限”标签内，点击账户公钥。在弹出的“私钥查看器”窗口内，点击“显示”。





## 什么是交易手续费

当您向API server发起交易请求时，您需要在交易内预先填写为该交易执行所支付的手续费，手续费费率可以通过api进行查看：<https://api.cybex.io/v1/refData>.

以下为一个交易手续费的参考示例：

"fees": {

"feeAssetId": "1.3.0",

"newFee": 55,

"cancelFee": 5,

"cancelAllFee": 50

}

交易手续费默认使用CYB支付(asset id: 1.3.0)，CYB的精度是5。以下为每种类型交易实际支付的手续费数量：

|  |  |  |
| --- | --- | --- |
| Transaction | Value | Actual cost |
| New | 55 | 55/(10^5) = 0.00055 CYB |
| Cancel | 5 | 5/(10^5) = 0.00005 CYB |
| Cancel All | 50 | 50/(10^5) = 0.0005 CYB |

## 更多相关信息

如果您需要了解更多与Cybex相关的信息，可以访问Cybex官方Wiki: https://github.com/CybexDex/cybex-node-doc/wiki

## 如何安装JDK或JRE

请访问 <https://adoptopenjdk.net/installation.html> 并选择您的操作系统，根据操作步骤完成安装。以下截图以Linux x64位为例：

