**Cybex API Server user manual**

Contents

[1 Overview 3](#_Toc536545402)

[2 Topology 3](#_Toc536545403)

[3 Cyb-signer 4](#_Toc536545404)

[3.1 Setup in Linux 4](#_Toc536545405)

[3.1.1 Prerequisite software 4](#_Toc536545406)

[3.1.2 Download cyb-signer source code 4](#_Toc536545407)

[3.1.3 Configure your cyb-signer 5](#_Toc536545408)

[3.1.4 Obtain artefact 5](#_Toc536545409)

[3.1.5 Start and stop cyb-signer 6](#_Toc536545410)

[3.1.6 Sign transaction 6](#_Toc536545411)

[3.2 Setup in Windows 8](#_Toc536545412)

[3.3 Setup in Mac 9](#_Toc536545413)

[4 API Server 9](#_Toc536545414)

[5 Frequently Asked Questions 17](#_Toc536545415)

[5.1 How to find my account name and account id 17](#_Toc536545416)

[5.2 How to find my private key 17](#_Toc536545417)

[5.3 What is the transaction fee 18](#_Toc536545418)

[5.4 General information 19](#_Toc536545419)

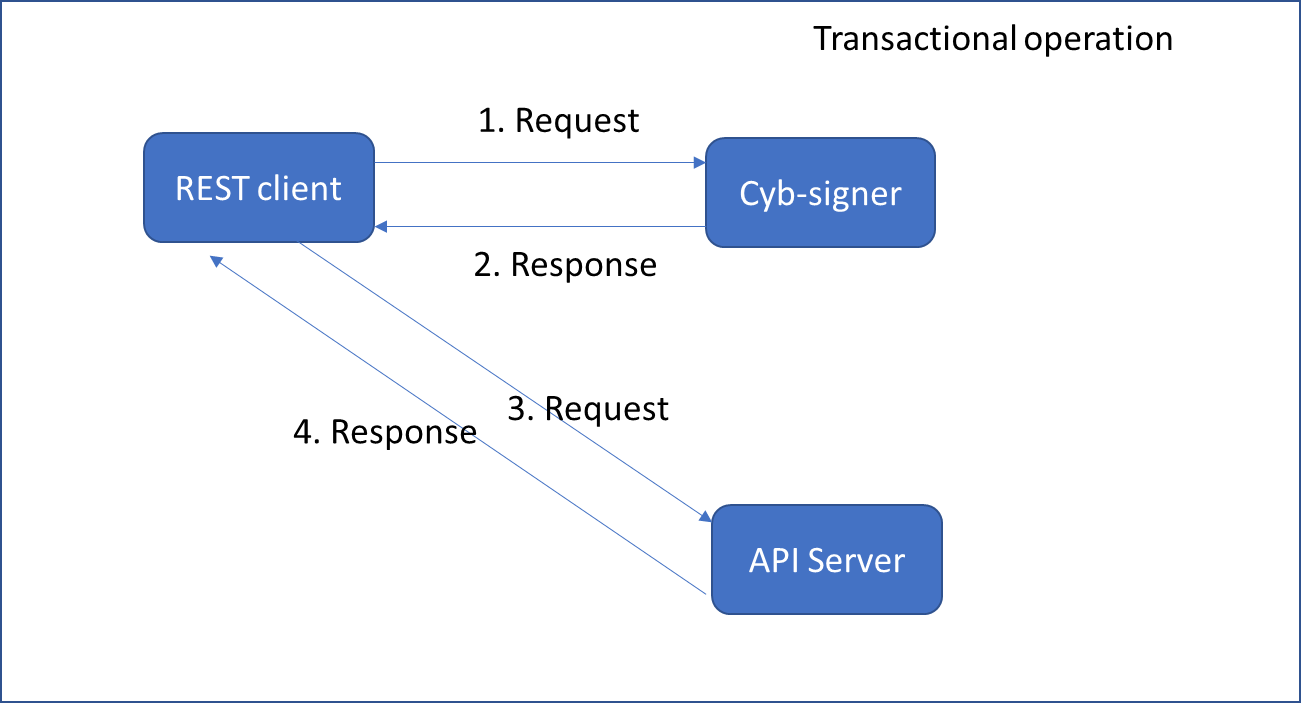
[5.5 How to install JDK or JRE 19](#_Toc536545420)

# Overview

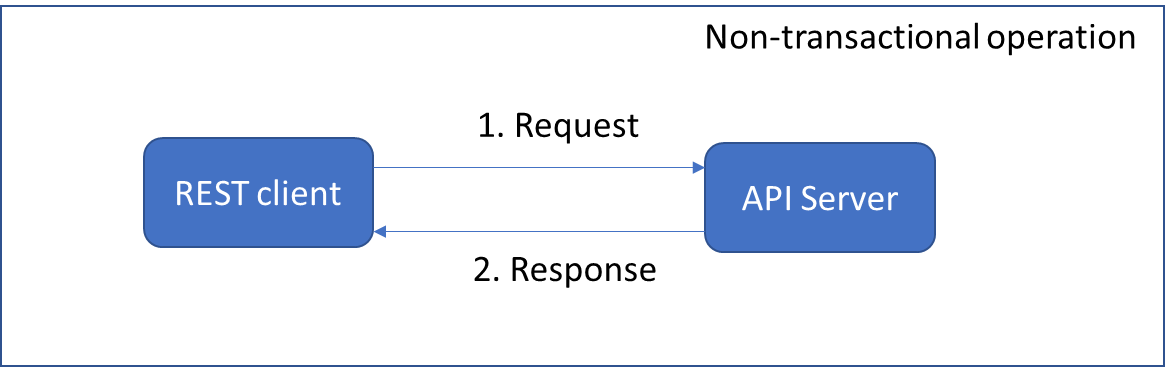
This document is to guide user to set up local cyb-signer and explain steps to use cybex API Server.

# Topology

The transactional operation is required to sign with user’s private key before sending to API server. User can use the response message from cyb-signer and send the exact content to API server.



However, non-transactional operation can be sent to API server directly via REST.

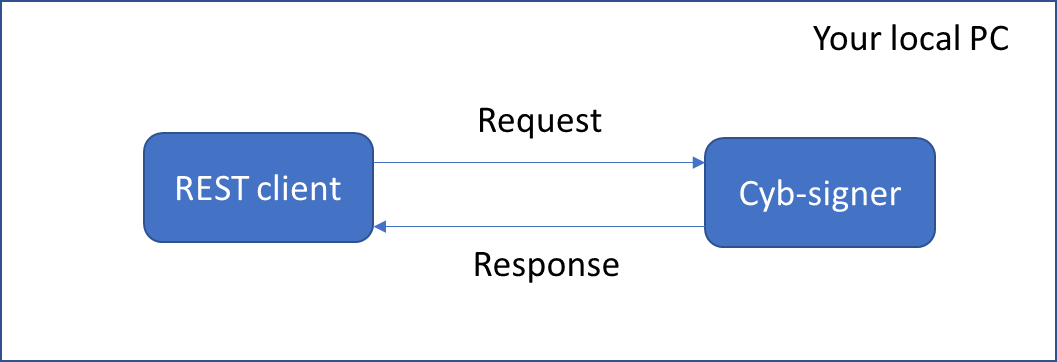


Following table lists all latest supported operations.

|  |  |
| --- | --- |
| **Type** | **Name** |
| Transactional | Create new order |
| Create cancel order |
| Create cancel-all order |
| Non-transactional | Query order status |
| Query trade status |
| Get reference data |
| Get user’s position |
| Get asset pair’s order book |
| Get asset pair’s candle stick info |

# Cyb-signer

This section is to guide user to set up their local cyb-signer step by step. The program is to create a REST server with user’s private key in your local computer. This could avoid expose user’s private key to other party while keeping the transaction signing process simple. User could use any REST client to sign a transaction easily.



## Setup in Linux

In following section, it will show you how to set up in Linux environment. All screen shots were captured from Ubuntu 16.04.

### Prerequisite software

Some software are required to install in order to proceed further.

|  |  |  |
| --- | --- | --- |
| Purpose | Software | Version |
| Use cyb-signer directly | JRE | 1.8 or later |
| Git | 2.7 or later |
| Build from source code and launch cyb-signer | JDK | 1.8 or later |
| Git | 2.7 or later |
| Maven | 3.0 or later |

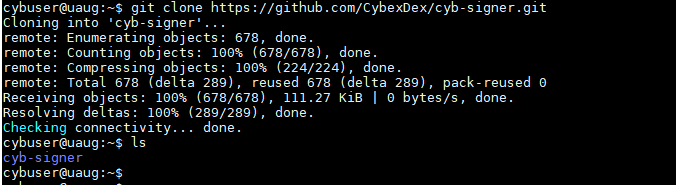
Please refer to section 5.5 on how to install JRE or JDK.

### Download cyb-signer source code

The full source code is in public github repository and mirror repository as well.

|  |  |  |
| --- | --- | --- |
| Type | Url | Note |
| Primary repository | Source code: https://github.com/CybexDex/cyb-signer.git |  |
| Release: https://github.com/CybexDex/cyb-signer/releases |  |
| Mirror repository | Source code: https://git.coding.net/david\_lu/cyb-signer.git | Suggested repo if connected from mainland China |
| Release: https://coding.net/u/david\_lu/p/cyb-signer/git/releases |

Please use “git clone” command to download it to your local PC.



### Configure your cyb-signer

All parameters present in the file env.properties under “cyb-signer/scripts” folder.

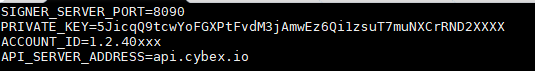
Following are example for your reference:

SIGNER\_SERVER\_PORT=8090 (an available port in your local pc)

PRIVATE\_KEY=5JicqQ9tcwYoFGXPtFvdM3jAmwEz6Qi1zsuT7muNXCrRND2XXXX (your private key)

ACCOUNT\_ID=1.2.xxxxx (your account id in cybex)

API\_SERVER\_ADDRESS=api.cybex.io (this is the same for everyone)



WARNINGS:

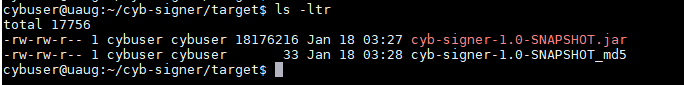
* ACCOUNT ID is not the same as ACCOUNT NAME and its format is 1.2.XXXXX. Please refer to section 5.1 for details.
* Private key is not your login password in cybex website. Please refer to section 5.2 below.

### Obtain artefact

You could either download the artefact from release url or build your artefact from source code.

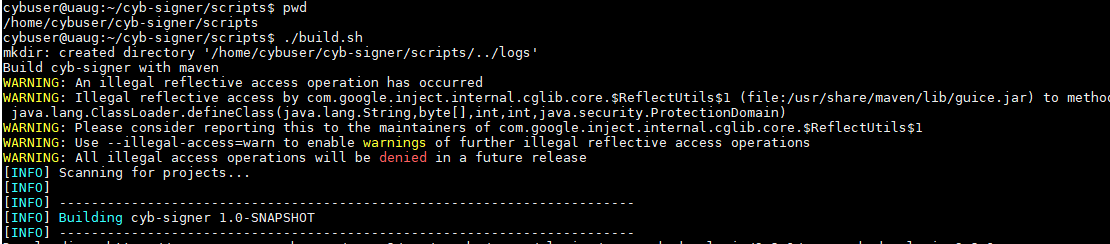
#### Download artefact from release url

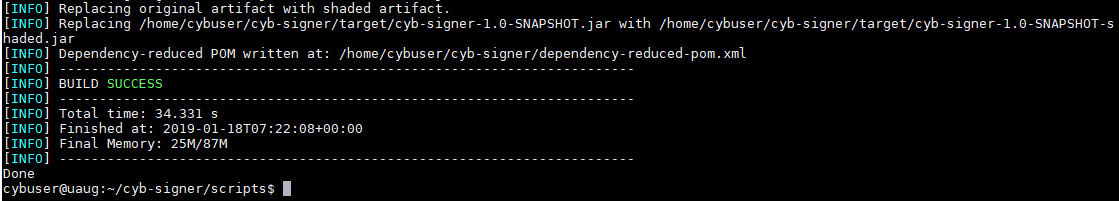
If you want to download the jar file, please access corresponding url stated in section 3.1.2 and put the jar file in “cyb-signer/target” folder.



#### Build your artefact from source code

If you prefer to build the jar file from source code, you could make it by execute “build.sh” under “cyb-signer/scripts” folder. When you build it for the first time, it takes a little longer since maven will download necessary artefacts.

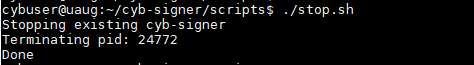




### Start and stop cyb-signer

You could use start.sh/stop.sh to launch/stop cyb-signer. Once cyb-signer starts, you don’t need to restart it unless you want to change parameters.





To increase security of signing, the cyb-signer and your REST-client must be in the same computer. cyb-signer will reject http request not using localhost in the url. For example, following request will fail: http://192.168.1.10:8090/signer/v1/newOrder.

### Sign transaction

Now you could use your REST client to send request message to cyb-signer to sign your transaction.

* Create New Order

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

|  |  |
| --- | --- |
| Parameter | Example value |
| assetPair | ETH/USDT, EOS/USDT |
| price | 146.67 |
| quantity | 10 |
| side | buy or sell |

|  |  |
| --- | --- |
| Example |  |
| Request | curl --data '{"assetPair": "ETH/USDT","price": 146.67,"quantity": 10,"side": "buy"}' http://localhost:8090/signer/v1/newOrder |
| Response | {  "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  } |

* Create Cancel Order

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

|  |  |
| --- | --- |
| Parameter | Example value |
| originalTransactionId | 88cecaa11b8584fb21243cd57ed2227e7c181452 |

|  |  |
| --- | --- |
| Example |  |
| Request | curl --data '{"originalTransactionId": "88cecaa11b8584fb21243cd57ed2227e7c181452"}' http://localhost:8090/signer/v1/cancelOrder |
| Response | {  "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"  } |

* Create CancelAll

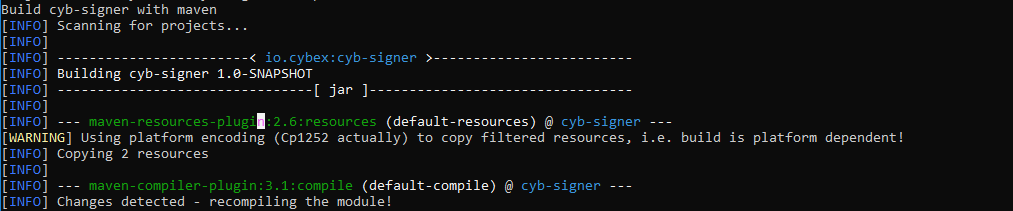
POST http://localhost:8090/signer/v1/cancelAll

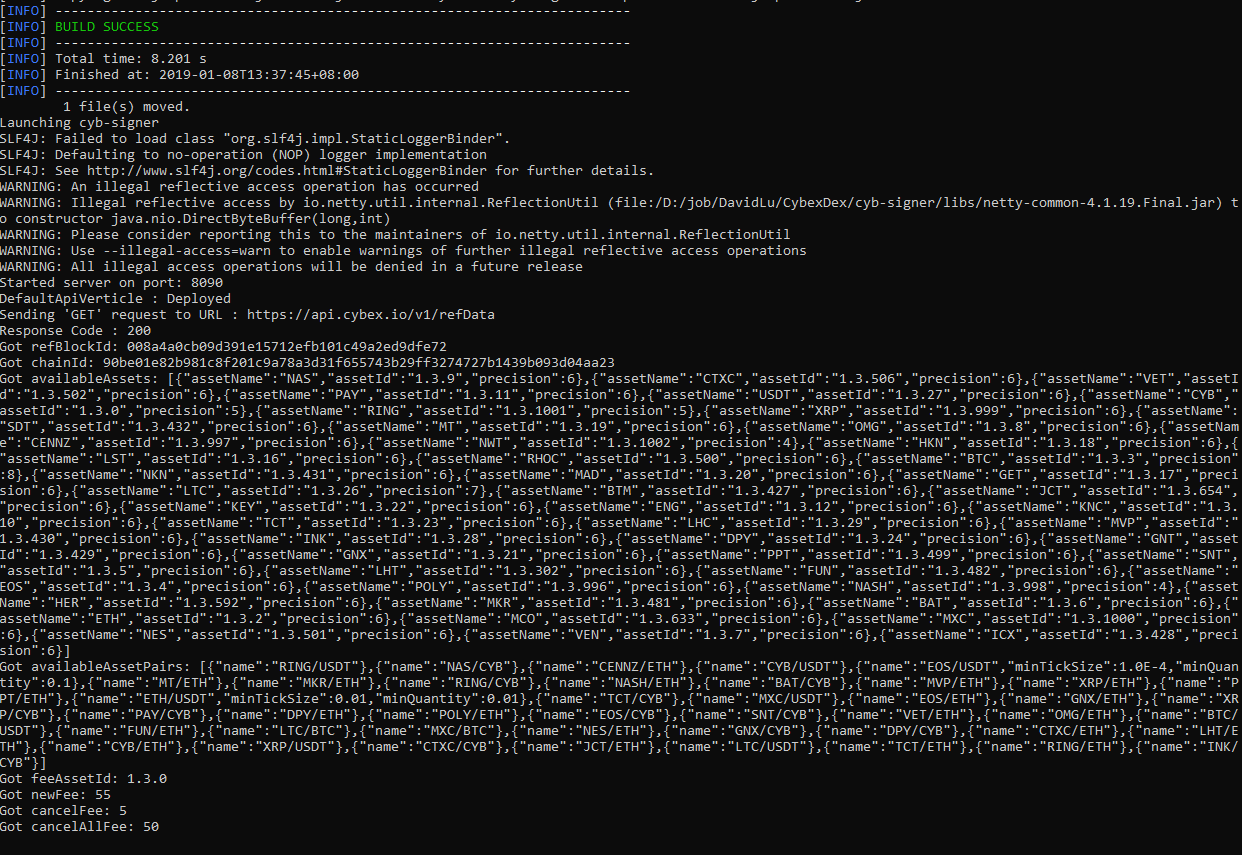
|  |  |
| --- | --- |
| Parameter | Example value |
| assetPair | ETH/USDT. Use “CYB/CYB” to cancel all your open orders no matter what asset pairs they are in. |

|  |  |
| --- | --- |
| Example |  |
| Request | curl --data '{"assetPair": "ETH/USDT"}' http://localhost:8090/signer/v1/cancelAll |
| Response | {  "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"  } |

## Setup in Windows

You could refer to 3.1 for most of setup except for the scripts in Windows. Once you configure in env.properties file, you could use “build.bat” to compile source code and use “start.bat” to launch it. When you want to shut it down, simply close the window.



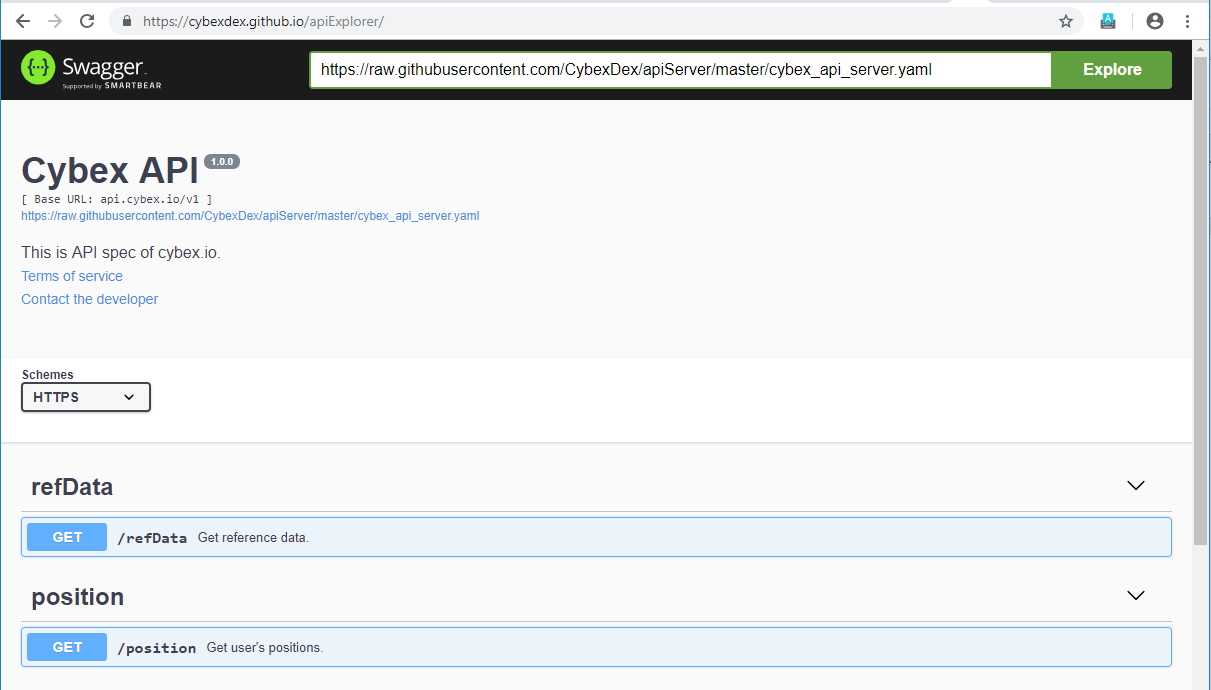


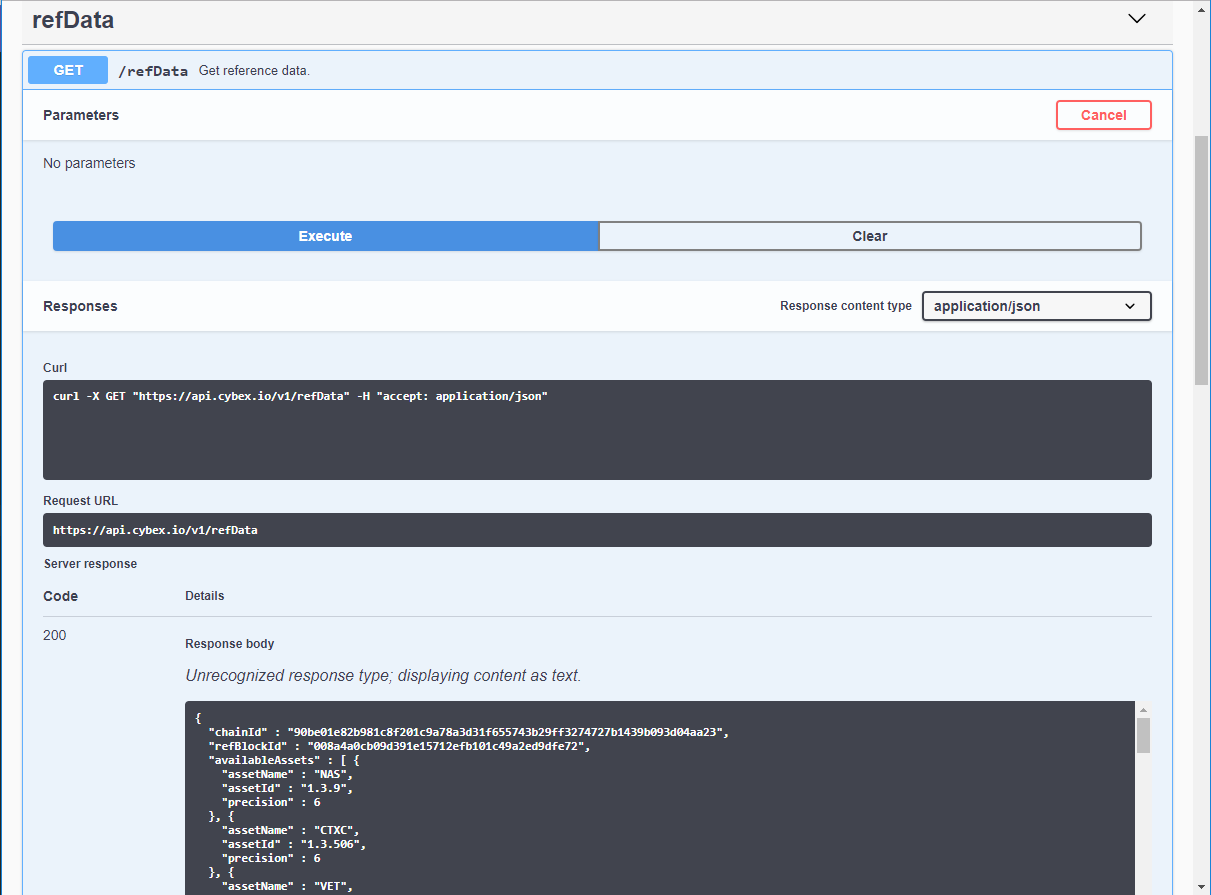
## Setup in Mac

You could refer to 3.1 for most of setup except for the scripts in Mac. Once you configure in env.properties file, you could use “build.sh” to compile source code and use “startMac.sh/stopMac.sh” to launch or stop it.

# API Server

As described in section 2, the API server is used to execute a lot of transactions and queries on behalf of users. User could navigate and have a test via api explorer: https://cybexdex.github.io/apiExplorer





In the next few pages, all API and examples will be illustrated.

* Query reference data

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

It will return some static data regarding exchange, e.g. chain id, reference block id, available assets, available asset pairs and fee info.

|  |  |
| --- | --- |
| Parameter | Example value |
| N/A |  |

|  |  |
| --- | --- |
| Example |  |
| Request | curl https://api.cybex.io/v1/refData |
| Response | {  "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  }  } |

* Execute transaction

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

User needs to put the return content from cyb-signer in the content of this post message.

|  |  |
| --- | --- |
| Parameter | Example value |
| Json content | The exact response from cyb-signer. It could be New, Cancel or CancelAll orders. |

|  |  |
| --- | --- |
| Example |  |
| Request | 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 |
| Positive Response | {  "Status": "Successful",  "orderSequence": 529016,  "signature": "1b477931ebb89b39f7e1f1953c9ead66f1c93eee8183f4334729e8f1  9dceab2a5000af53728952dd73bc3636ab0663cc8f0293714d4c4ed8f3399c5a50b9  da2816",  "time": "2019-01-08T05:53:19.049071Z"  } |
| Negative Response | {  "Status" : "Failed",  "Code" : 1005,  "Message" : "Duplicated transaction",  "signature" : "1b477931ebb89b39f7e1f1953c9ead66f1c93eee8183f4334729e8f19dceab2a  5000af53728952dd73bc3636ab0663cc8f0293714d4c4ed8f3399c5a50b9da28  16"  } |

* Query order status

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

Use this api to query order statuses.

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Mandatory | Example value |
| accountName | STRING | YES | Your-account-name |
| reverse | BOOL | NO | If it is true, the most recent record will be shown at the beginning. By default, it is true. |
| assetPair | STRING | NO | Asset pair filter. E.g. ETH/USDT |
| orderStatus | STRING | NO | Order status filter. E.g. FILLED. Use comma as delimiter to include more than one statuses. E.g. FILLED, OPEN |
| startTime | STRING | NO | Beginning time of your query, format must be YYYY-MM-DDTHH:mm:ss.ssssssZ. E.g. 2018-01-07T01:20:48.647910Z |
| endTime | STRING | NO | End time of your query, format must be YYYY-MM-DDTHH:mm:ss.ssssssZ. E.g. 2019-01-07T01:20:48.647910Z |
| start | INT | NO | Number where result will be shown from |
| limit | INT | NO | Max number of results returned |
| transactionId | STRING | NO | Query by transaction id |

|  |  |
| --- | --- |
| Example |  |
| Request | curl 'https://api.cybex.io/v1/order?accountName=XXXX&reverse=true&orderStatus=FILLED,CANCELED' |
| Positive Response | [ {  "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"  }] |

* Query trade

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

Use this api to query trades.

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Mandatory | Example value |
| accountName | STRING | YES | Your-account-name |
| reverse | BOOL | NO | If it is true, the most recent record will be shown at the beginning. By default, it is true. |
| assetPair | STRING | NO | Asset pair filter. E.g. ETH/USDT |
| startTime | STRING | NO | Beginning time of your query, format must be YYYY-MM-DDTHH:mm:ss.ssssssZ. E.g. 2018-01-07T01:20:48.647910Z |
| endTime | STRING | NO | End time of your query, format must be YYYY-MM-DDTHH:mm:ss.ssssssZ. E.g. 2019-01-07T01:20:48.647910Z |
| start | INT | NO | Number where result will be shown from |
| limit | INT | NO | Max number of results returned |

|  |  |
| --- | --- |
| Example |  |
| Request | curl 'https://api.cybex.io/v1/trade?accountName=XXXXXX&reverse=true' |
| Positive Response | [ {  "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"  } ] |

* Query user’s position

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

User uses it to query his current position.

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Mandatory | Example value |
| accountName | STRING | YES | Your-account-name |

|  |  |
| --- | --- |
| Example |  |
| Request | curl 'https://api.cybex.io/v1/position?accountName=XXXX' |
| Positive Response | {  "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"  } |

* Query order book for an asset pair

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

User uses it query order book.

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Mandatory | Example value |
| assetPair | STRING | YES | E.g. ETH/USDT, EOS/USDT |
| limit | INT | YES | Number of levels |

|  |  |
| --- | --- |
| Example |  |
| Request | curl 'https://api.cybex.io/v1/orderBook?assetPair=ETH/USDT&limit=3' |
| Positive Response | {  "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"  } |

* Query candle stick

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

Use it to query asset pair’s candle stick.

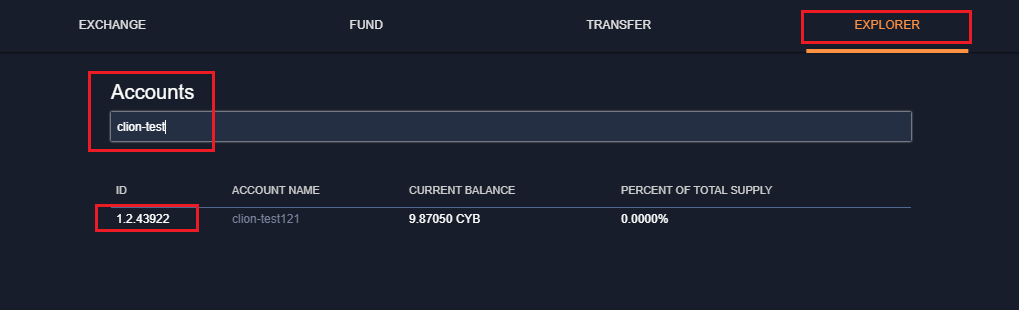
|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Mandatory | Example value |
| assetPair | STRING | YES | E.g. ETH/USDT, EOS/USDT |
| interval | ENUM | YES | 1m, 3m, , 5m, 15m, 30m, 1h, 2h, 4h, 6h, 8h, 12h, 1d, 3d, 1w, 1M |
| startTime | LONG | NO | Beginning time of the query in epoch time. |
| endTime | LONG | NO | End time of the query in epoch time. |
| limit | INT | NO | Level to be shown |

|  |  |
| --- | --- |
| Example |  |
| Request | curl 'https://api.cybex.io/v1/klines?assetPair=EOS/USDT&interval=1m&limit=2' |
| Positive Response | [  [  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"  ]  ] |

# Frequently Asked Questions

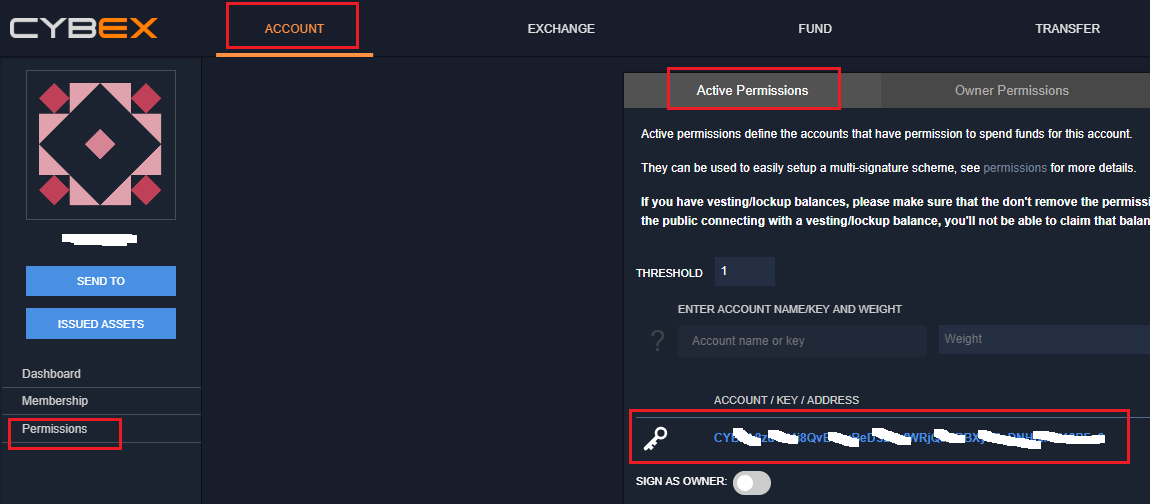
## How to find my account name and account id

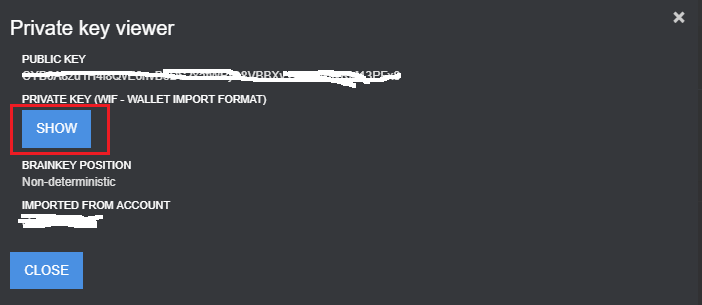
Account name is the user name which you register in cybex.io. For account id, please first log in https://olddex.cybex.io. Under Explorer->Accounts, please input your account name, then you will find your account id below.



## How to find my private key

Please log in https://olddex.cybex.io, under Account->Permissions->Active Permissions, when you click the address then click “SHOW” in the pop-up page, your private key will be shown.





## What is the transaction fee

When you instruct API server to execute a transaction, a pre-filled fee is included in your transaction message. The detail fee schedule could be found in this api: https://api.cybex.io/v1/refData.

There is a section regarding fee. The numbers below are for reference only and subject of change.

"fees": {

"feeAssetId": "1.3.0",

"newFee": 55,

"cancelFee": 5,

"cancelAllFee": 50

}

By default, fee is deducted as CYB (asset id: 1.3.0). CYB’s precision is 5. The actual cost per transaction could be calculated as follows:

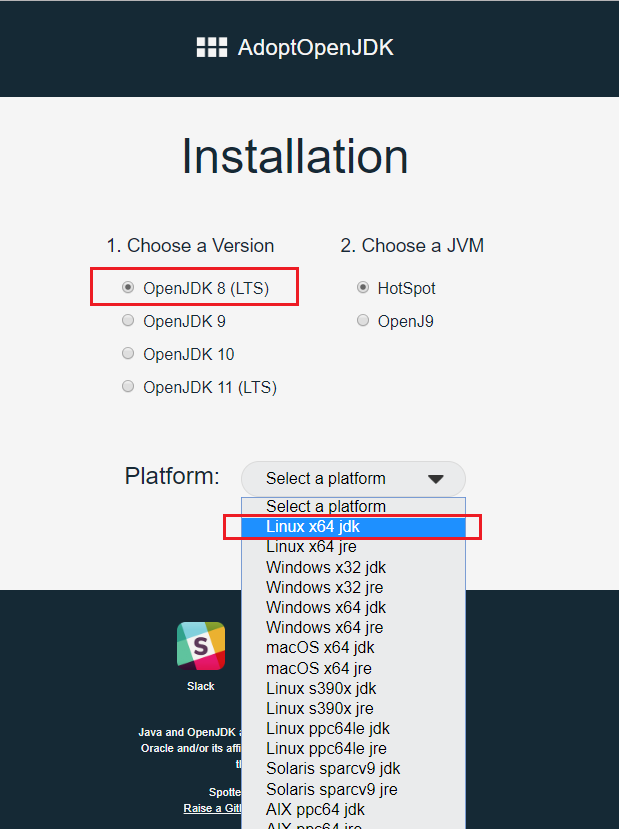
|  |  |  |
| --- | --- | --- |
| Transaction | Value | Actual cost |
| New | 55 | 55/(10^5) = 0.00055 CYB |
| Cancel | 5 | 5/(10^5) = 0.00005 CYB |
| Cancel All | 50 | 50/(10^5) = 0.0005 CYB |

## General information

Please refer to cybex wiki page for general information: https://github.com/CybexDex/cybex-node-doc/wiki

## How to install JDK or JRE

1. Please access https://adoptopenjdk.net/installation.html
2. Select your OS platform. Following uses linux x64 as example.



You could easily switch to Windows or macOS.

1. Please follow instruction to complete the setup

