
Read the Docs Template Documentation

Release 1.0

Read the Docs

Feb 26, 2019

1	Introduction to YOYOW	1
1.1	YOYOW Role Conception	1
1.2	Private Key System	2
1.3	Types of Assets	2
1.4	White Paper	3
2	Testnet	5
2.1	Executable Files for Testnet	5
2.2	Testnet Web Wallet	5
2.3	Running the Testnet Node	5
2.4	Testnet Explorer	5
2.5	Testnet Public Node	5
2.6	Running Testnet Wallet	6
2.7	Others	6
3	User Registration	7
3.1	Registering Accounts	7
3.2	Private Key Authority	7
3.3	Private Key Saving	8
4	Compiling and Running yoyow-core	9
4.1	Recommended Hardware Environment	9
4.2	Installation Dependence:	9
4.3	Compile Script:	9
4.4	Running:	9
5	Introduction to the Platform	11
5.1	Platform Requirements	11
5.2	Users Authorizing the Platform	11
6	Platform Related Operations	13
6.1	Getting Platform Number	13
6.2	Getting Platform Info	13
6.3	Checking Platform List	13
6.4	Creating Platforms	14
6.5	Undating Platform Info	14
6.6	Voting for Platforms	14

6.7	Authorizing Platforms	14
6.8	Canceling the Platform Authorization	14
7	Introduction to Witness	15
7.1	Functions of Witness	15
7.2	Witness Qualification (Collateral)	15
7.3	Ways to Become Witnessess	15
7.4	Rewards of Witnesses	16
7.5	Witness Election	16
7.6	Penalty Mechanism for Witnesses	16
7.7	Witnesses being Offline	16
7.8	Witness Related FAQ	17
8	Witness Operations	19
8.1	Witness Node Construction	19
8.2	Query	19
8.3	Creating Witnesses	19
8.4	Updating the Info of Witnesses	20
8.5	Witness Voting	20
8.6	Witness being Offline	20
8.7	Getting Witness Returns	20
9	Introduction to Committee	21
9.1	The Responsibilities of Committee	21
9.2	The Committee Setup	21
9.3	Candidates for the Committee	21
9.4	Committee Elections	22
9.5	Committee Proposals	22
9.6	Resignation of Committee Members	22
10	Committee Related Operations	25
10.1	Creating Committee Account	25
10.2	Updating Committee Account	25
10.3	Voting for Committee	25
10.4	Getting Info of Committee Members	26
10.5	Checking History of Current Candidates	26
10.6	Checking the Current Committee Proposals Not Implemented	26
11	Voting	27
11.1	Voting Qualification	27
11.2	Validity Period of Voting	27
11.3	Voting Method	27
11.4	Valid Vote Counting	28
11.5	Voting Proxies	28
12	Collateral	29
12.1	Platform Collateral	29
12.2	Committee Collateral	29
12.3	Miner Witness Collateral	29
12.4	Collateral Refunding	30
13	YOYO Token	31
13.1	Balance	31
13.2	Tipping	31
13.3	The Design of Balance and Tipping	31

14	User-Issued Assets (UIA)	33
14.1	Issuing Assets	33
14.2	Advanced Options	34
15	Bonus Points	35
15.1	Using Bonus Points	35
15.2	Acquisition of Bonus Points	35
15.3	Renting Bonus Points	35
15.4	Gifting Bonus Points	35
16	Asset Related Operations	37
16.1	Common Transfer	37
16.2	Query	37
16.3	About Bonus Points	38
16.4	User-Issued Assets	38
17	Node API Descriptions	41
17.1	Node Connection Method	42
17.2	API Classification	42
17.3	Database API	43
17.4	History API	77
18	Wallet API Descriptions	81
18.1	Wallet Connection Method	83
18.2	2.1 Tools API	84
18.3	2.2 Queries API	87
18.4	2.3 Wallet/Private Key Management API	116
18.5	2.4 Operations/Transactions API	124
19	YOYOW Middleware InstructionNodejs	167
19.1	Deployment Start	167
19.2	Interface Descriptions	169
20	Descriptions of YOYOW Parameter Table and Fees	183
21	Instructions for Exchanges	185
21.1	Preparation	185
21.2	Installation	186
21.3	Starting YOYOW Node	186
21.4	Starting YOYOW Client	187
21.5	Accessing the Client via HTTP-RPC	191
21.6	Processing Deposits	191
21.7	Process Withdrawal Requests	193
21.8	Sample Code	194
22	Introduction to Platform Integration	195
22.1	Introduction to YOYOW Ecology	195
22.2	Integration Process	198
23	Creating YOYOW Platform Account from scratch	203
23.1	1. Creating Accounts, Getting Private Keys	203
23.2	2. Creating Platforms	205
24	More Materials	209
25	Indices and tables	211

Introduction to YOYOW

YOYOW is named from “You Own Your Own Words”. Its goal is to build a network that applies the blockchain technology and uses decentralized consensus method to price contribution and provide equity returns for the content area, and to build a reasonable content value distribution mechanism, which enables content producers, content investors, content filterers and eco-builders to get reasonable incentives and rewards while building a value network based on users’ content ratings.

Unlike other blockchain systems, YOYOW has designed more user roles, unique key systems and rich asset types by using DPoS consensus mechanism in order to adapt to the ecology of content production.

1.1 YOYOW Role Conception

In the system ecology, in order to balance efficiency and fairness, YOYOW has designed a complete user system, including common users, platforms, committees, and witnesses.

1.1.1 Common Users

Common users are the main participants of the YOYOW network. In the YOYOW network, common users share a unified account system on each platform, and have the rights of transfer, election, etc.

1.1.2 Platforms

Common user that deposits a collateral of a certain amount of YOYO tokens can be called a platform. Common users can authorize the platform through the YOYOW wallet, and the authorized platform uses its tipping authority for cross-site login, content rating, content posting, comments, etc., and common users can also revoke authorization at any time.

1.1.3 Committee

The committee is the governing body of the YOYOW network voted in by the YOYOW users and is responsible for initiating and voting on the proposals. The content of the proposals is mainly to adjust the adjustable parameters of the system, such as the rate of various transactions, the highest scoring weight, etc., including as well the block production interval, block rewards, etc.

1.1.4 Witnesses

The witness is the producer of the block, responsible for collecting the various transactions of the broadcast and packing them into the blocks, so the witnesses producing blocks can also get the corresponding YOYO token as returns.

1.2 Private Key System

In the blockchain project, the private key is the top priority of the account. However, when a user does any operation and uses the account private key to sign, it will undoubtedly greatly increase the risk of the account. Considering the different security levels of different authorities, the three-level key system is set in the YOYOW system: **the Owner Key**, **the Active Key**, and **the Secondary Key**.

- The Owner Key is the key of highest authority, and the Owner Key should not be used unless it is absolutely necessary;
- The Active Key can operate the balance;
- The Secondary Key can operate the tipping, and can also be used for posting, giving likes, login authentication, etc.
- In addition to the asset-related 3-level keys, there is also a **Memo Key** used to encrypt and view the note information for the transactions.

1.3 Types of Assets

There are three types of assets in the YOYOW network: **YOYO token**, **user-issued assets (UIA)**, and **bonus points**.

1.3.1 YOYO Token

In YOYO's account, the base asset is YOYO token, and there are two places to store: **balance and tipping**

- **Balance** has a high level of security, and it is recommended that a large amount of YOYO be stored in the balance.
- **Tipping** is for small amount of savings and transfer. When the user grants the authority of tipping to the platform, the platform can have the right to operate "tipping", which can be understood as password-free payment.

1.3.2 User-Issued Assets UIA

Users can issue custom assets: users can create Tokens that circulate within their own platforms based on their business. A convenient exchange method between the user-issued assets (UIA) and YOYO will be provided subsequently, sharing convenient monetization channels.

1.3.3 Bonus Points

The function of the bonus points is to pay fees. When the account “balance” has a certain amount of YOYO, the system will automatically accumulate points. When paying fees, you can choose to deduct the points. For more information, please refer to: [Bonus Points](#)

1.4 White Paper

For the content of white paper, please refer to [White Paper Download](#)

2.1 Executable Files for Testnet

The executable files for testnet can be downloaded on [yoyow-core-testnet](#)

2.2 Testnet Web Wallet

The website address of the web wallet used in testnet: <http://demo.yoyow.org:8000/#/>

You can register your testnet account in the web wallet. After registration, you will receive 12000 TEST YOYO.

2.3 Running the Testnet Node

Set up a testnet node, run the downloaded `yoyow_node` of the testnet directly, taking the Ubuntu environment as an example.

```
./yoyow_node --rpc-endpoint 127.0.0.1:8090
```

2.4 Testnet Explorer

Testnet Explorer: <https://testnet.explorer.yoyow.org/>

2.5 Testnet Public Node

If you don't want to build a testnet full node by yourself, you can use the public full node directly:

```
websocket interface address: ws://47.52.155.181:10011  
jsonrpc interface address: http://47.52.155.181:10011/rpc
```

2.6 Running Testnet Wallet

If using public nodeyoyow_client needs to be added `-s` parameter to specify the node address of the link

```
./yoyow_client -s ws://47.52.155.181:10011 -r 0.0.0.0:8091 -H 127.0.0.1:8093
```

2.7 Others

If you encounter any problems in testing, such as insufficient test tokens, please add the WeChat of the assistant: yoyow99; apply for the membership of the YOYOW developer group.



3.1 Registering Accounts

The account of the YOYOW platform is a string of numbers (similar to the QQ number). It is assigned by the system when registering, and the nickname function will be released in the future.

Currently, accounts can only be registered through [Web Wallet](#) or [Client Wallet](#). You can refer to: [Web Wallet Registration Method](#)

Special attention is needed here. The system will generate a private Owner Key when registering. The private key only appears once upon successful registration. Please make sure to back up and save it properly since the web wallet and client wallet will not store your private Owner Key.

3.2 Private Key Authority

In the YOYOW system, there are four types of private keys for each account.

- **Owner Key:** it has the highest authority. It is the private key that can modify other authorities of the account. The private key that appears upon successful registration is the Owner Key.
- **Active Key:** it has the authority to use the assets in the balance, such as controlling the actions of the transfer.
- **Secondary Key:** it has the authority to use the tipping, including the actions of authorizing the platforms. It is also called tipping Key.
- **Memo Key:** it is used to check the memo information of the transfer actions.

Authorization at each level is for actions within that level only. No action beyond or below one's authorization is allowed. Active Key does not have the right to transfer tipping, and Secondary Key does not have the right to transfer balances either. The private Owner Key does not have the authority to use balances and tipping either, but it can reset the private keys of other authority levels. Other three types of private keys can be reset by importing the Owner Key of the account in the web wallet.

For more info about private keys, you can refer to [yoyow private key tutorials](#)

3.3 Private Key Saving

Private Owner Key only appears once upon successful registration. The browser will not save the private key, and neither will YOYOW. It is recommended to save your private Owner Key properly and safely and ensure that it is free from being lost and stolen. It is dangerous to store the private key in plaintext on netdisk or online notes. It is also recommended not to use private Owner Key easily.

In the web wallet, other private keys from lower authority levels will be encrypted using your wallet password and stored in the browser's local database. The same is true for the client wallet. Your account can be rebuilt using the private Owner Key while certain circumstances (such as switching computer, computer failure or browser crash) lead to the loss of local data on your browser.

Compiling and Running yoyow-core

4.1 Recommended Hardware Environment

System configuration: two cores, 4G memory

Recommended System: Ubuntu 16.04 LTS (64-bit)

4.2 Installation Dependence:

```
sudo apt-get update
sudo apt-get install autoconf cmake make automake libtool git libboost-all-dev libssl-
↳dev g++ libcurl4-openssl-dev
```

4.3 Compile Script:

```
git clone https://github.com/yoyow-org/yoyow-core.git
cd yoyow-core
git checkout yy-mainnet # may substitute "yy-mainnet" with current release tag
git submodule update --init --recursive
mkdir build
cd build
cmake -DCMAKE_BUILD_TYPE=Release ../
make yoyow_node
make yoyow_client
```

4.4 Running:

After the compilation is completed, two binary files are generated, `yoyow_node` and `yoyow_client`.

Yoyow_node is generally called a node program and is mainly used to connect to a blockchain network. The functions include synchronizing data on the chain, broadcasting transactions, and providing basic [API](#), etc.

Yoyow_client is generally called a client program. It further encapsulates the API provided by the node program to provide users with a more friendly API. It also provides private key storage management and cryptographic signature function, so it is also called a wallet program.

4.4.1 Node Program

```
./programs/yoyow_node/yoyow_node
```

When the node program runs, it automatically creates a directory `witness_node_data_dir` that stores data and configuration files. It takes several hours to synchronize all the data. After the synchronization is completed, you can stop the operation with `Ctrl+C`.

It should be noted that after inputting `Ctrl+C`, the program will continue to run for a while. Do not type `Ctrl+C` again. Otherwise, the database will be inconsistent. It may take a long time to replay the next time you start the node.

After receiving `Ctrl+C`, the program will print out in the log

```
1553960ms asio      main.cpp:238      operator()      ] Caught_
↪SIGINT attempting to exit cleanly
1553960ms th_a     main.cpp:251     main           ] Exiting_
↪from signal 2
```

At this point, wait patiently for the program to exit.

When the node program runs, the websocket interface is not opened by default, and you need to add parameters when executing.

```
./programs/yoyow_node/yoyow_node --rpc-endpoint 127.0.0.1:8090
```

You can also directly modify the configuration of `rpc-endpoint` in the configuration file `witness_node_data_dir/config.ini`.

4.4.2 Client Program (Wallet)

The client program needs to connect to the websocket interface of the node program at runtime. By default, it will try to connect to `127.0.0.1:8090`. You can also use the `-s` parameter to specify other addresses.

```
./programs/yoyow_client/yoyow_client -s ws://47.52.155.181:10011
```

The client program provides a command line that can be interactive. You can perform some operations by typing the relevant commands through the command line.

As mentioned above, the client program saves the private key. For security reasons, in the operation of using the private key, you need to use the password to unlock the client. The password settings and unlock command are as follows:

```
>>> set_password <PASSWORD>
>>> unlock <PASSWORD>
```

Then you can import the private key

```
unlocked >>> import_key <account> <private_key>
```

More supported commands can be viewed with the help command or by looking at the [wallet api](#) documentation.

Introduction to the Platform

The platform is a service provider that provides content storage, user interaction, authorized login and other services in the YOYOW network. The third party can become a platform through the collateral of YOYO tokens, and the committee has the right to revoke the qualification of the platform and impose other penalties when the platform violates the community consensus/regulation.

5.1 Platform Requirements

Common accounts need to deposit sufficient collateral to become platform accounts. Currently, 10,000 YOYO is required for collateral. The creation fee of 1000 YOYO is required to create the platform.

5.2 Users Authorizing the Platform

Common users can authorize the platform through the YOYOW wallet, and can authorize the platform proxies to use its authentication authority, which can be used for cross-site login, single sign-on, etc., sharing all users in the YOYOW ecosystem.

User authorization essentially means that the common account grants the authority of the tipping to the platform, and the common account can carry out small amount “password-free payment” in the ecosystem constructed by the platform, helping the platform to quickly have Token of strong circulation. The platform can also use its own issued Token. Please refer to [UIA](#).

More information about user authorization can be found in [ForPlatforms](#).

Platform Related Operations

The following related operation commands can be executed in the interactive command line in `yoyow_client`

6.1 Getting Platform Number

```
get_platform_count
```

For details, please refer to: `get_platform_count`

6.2 Getting Platform Info

```
# get_platform <owner_account>  
get_platform 250926091
```

For details, please refer to: `get_platform`

6.3 Checking Platform List

```
# list_platforms <lowerbound> <limit> <order_by>  
list_platforms 0 5 1
```

For details, please refer to: `list_committee_members`

6.4 Creating Platforms

```
# create_platform <owner_account> <name> <pledge_amount> <pledge_asset_symbol> <url>  
→<extra_data> <broadcast>  
create_platform 223331844 yoyow.club 10000 YOYO "http://yoyow.club" null true
```

For details, please refer to: `create_platform`

6.5 Updating Platform Info

```
# update_platform <platform_account> <name> <pledge_amount> <pledge_asset_symbol>  
→<url> <extra_data> <broadcast>  
update_platform 223331844 NUUUU null null "http://www.example.com" "http://www.  
→example.com" true
```

For details, please refer to: `update_platform`

6.6 Voting for Platforms

```
# update_platform_votes <voting_account> <platforms_to_add> <platforms_to_remove>  
→<broadcast>  
update_platform_votes 250926091 ["223331844"] [] true
```

For details, please refer to: `update_platform_votes`

6.7 Authorizing Platforms

```
# account_auth_platform <account> <platform_owner> <broadcast>  
account_auth_platform 250926091 223331844 true
```

For details, please refer to: `account_auth_platform`

6.8 Canceling the Platform Authorization

```
# account_cancel_auth_platform <account> <platform_owner> <broadcast>  
account_cancel_auth_platform 250926091 223331844 true
```

For details, please refer to: `account_cancel_auth_platform`

7.1 Functions of Witness

YOYOW uses a consensus mechanism for flexible selection of DPoS + PoS, and witnesses act as “miners” in traditional blockchain projects.

Its core responsibility is packaging transactions (commonly known as “producing blocks”) to maintain a stable network operation; witnesses enjoy the rewards given by the system.

7.2 Witness Qualification (Collateral)

Any account with collateral over a certain amount of YOYO can become a witness. Current minimum value for collateral: 10,000 YOYO. Witnesses are free to adjust the amount of collateral at any time.

The operation of witness registration costs 1000 YOYO, and the minimum value for collateral is 10000 YOYO, so at least 11,000 YOYO are required to become witnesses.

7.3 Ways to Become Witnesses

There are two ways to become witnesses of YOYOW:

1. voting election
2. YOYOW collateral

There are three types of witnesses in the above two ways:

1. Top witnesses: directly voted in by token holders, ranked according to the number of “effective votes”, being top 11.

2. Standby witnesses: directly voted in by token holders, ranked according to the number of “valid votes” and all other candidates apart from the top 11 are weighted according to the number of votes, taking turns to work alongside the top witnesses. Currently, three candidates ranked 12th to 14th are selected in each round.
3. Miner witnesses: generated by collateral ranking, weighted according to “valid collateral amount”, taking turns to work. For example, if A, B and C all deposit collateral, and A has six tokens of valid collateral, B has three tokens of valid collateral, and C has one token of valid collateral, then, for the following 100 blocks, it should be arranged that A produces 60 blocks, B produces 30 blocks and C produces 10 blocks. Seven miner witnesses are selected in each round.

To sum up: YOYOW’s witness generation mechanism is based on the “number of votes” and “collateral number”, pre-selecting 21 witnesses (11 + 3 + 7). The selected witnesses take turns to producing blocks, and the round of time is about 63 seconds. When 21 witnesses finish producing blocks, the next round of elections will be automatically carried out.

7.4 Rewards of Witnesses

Witnesses will get corresponding rewards every time a block is produced.

According to the resolution of the current committee:

- Top witnesses: 0.3 YOYO for each block
- Standby witnesses: 0.3 YOYO for each block
- Miner witnesses: 0.5 YOYO for each block

Witnesses can withdraw “available” rewards into their balance at any time.

7.5 Witness Election

- Top witnesses and standby witnesses are elected according to the ranking of votes. For detailed voting rules please refer to [Voting](#)
- Miner witnesses are elected according to the ranking of collateral. For collateral ranking rules, please refer to [Collateral](#)

7.6 Penalty Mechanism for Witnesses

- Under suspicion of an attempt to double spend: If a witness has two blocks at the same height, it is suspected of attempting to create forks and implement a double spend attack.
- If the network detects this situation, or if someone reports it, after the evidence is recorded, the witness is forced to go offline and some deposits are deducted.
- No rewards for reporting for the time being

7.7 Witnesses being Offline

- Go Offline on one’s ownwitnesses can set up a temporary offline status on their own when temporary maintenance is required.

- Go offline forcibly: When the witnesses lose blocks, if the last time of the block production is earlier than 24 hours before, the witness is automatically set to be temporarily offline with no further block production arranged, and the position is replaced by the witness of a lower rank.
- Witnesses can be arranged in the later block production if the witnesses reset an online status manually.

7.8 Witness Related FAQ

To be supplemented

8.1 Witness Node Construction

Complete tutorials for building witnesses can be found in [YOYOW Witness Tutorials \(Ubuntu\)](#)

8.2 Query

8.2.1 Getting Info of Witnesses

```
# get_witness <owner_account>  
get_witness 132826789
```

For details, please refer to: `get_witness`

8.2.2 Query for Witness List

```
# list_witnesses <lower_bound> <limit> <order_by>  
list_witnesses 0 5 1
```

For details, please refer to: `list_witnesses`

8.3 Creating Witnesses

```
# create_witness <owner_account> <block_signing_key> <pledge_amount> <pledge_asset_  
→symbol> <url> <broadcast>  
create_witness 223331844 YYW1111111111111111111111111111111114T1Anm 1000000 YOYO  
→"http://www.yoyow.org" true
```

Introduction to Committee

A group of committee members voted in by the token holders are responsible for setting parameters of the system operation.

9.1 The Responsibilities of Committee

- Adjusting the various system operation parameters, including fees and block production rewards
- The committee has partial decision-making authority

9.2 The Committee Setup

- The number of committee members is initially fixed at five, and cannot be modified.
- The tenure of committee members is set at 30 days, and cannot be modified.
- The committee members on the initial list will be appointed by the system, with subsequent committee members decided by votes taken by token holders.
- No remuneration for committee accounts

9.3 Candidates for the Committee

- Any account with collateral over a certain amount of YOYO can become a candidate for the committee. The initial figure is set at 1,000 YOYO.
- Only candidates for the committee are qualified to become formal committee members.

9.4 Committee Elections

- Each token holder has one vote that can be used for one candidate—that is to say, an account can only support one candidate at a time, and the voter can change their vote at any time.
- Other rules such as voting qualification, total valid votes, validity period for voting, and proxies are used in conjunction with witness voting. Please refer to [Voting](#)
- The five candidates with the most votes at re-election time will become the new incumbent committee members. Currently, there is no regulation on minimum votes.

9.5 Committee Proposals

- Incumbent committee members can initiate proposals to be voted on by all committee members.
- Each proposal includes the following attributes: proposal content, voting deadline, threshold for voting approval and the date from which it would take effect.
- The threshold for approval will be set in advance according to the type of proposal, and cannot be assigned. Only a threshold for approval has been set for the time being, with no threshold for disapproval.
- A proposal may contain more than one item of content, in which case approval or disapproval of all items is required, and the highest threshold for approval among the various content will be recognized as the threshold for approval.
- The date from which a proposal would take effect must not be later than the end date of the tenure. The date of effect can be backdated, resulting in it “taking immediate effect.”
- There are three statuses for committee votes—approving, abstaining and disapproving. It is assumed that the initiator of the proposal has voted to approve it. Each committee member shall have one vote, which can be changed before it comes into effect.
- Under the condition that the number of affirmative votes has reached the threshold, if the date of effect is in the past, the proposal will come into effect immediately. Otherwise, the proposal will come into force on the planned date of effect.
- Before the close of voting, and before the proposal’s date of effect, voters are allowed to change their votes. However, after the close of voting, no change will be accepted.
- Committee proposals shall be voted on within the tenure. Proposals whose voting has not finished within that time period will be automatically canceled upon the termination of the tenure. Approved proposals will come into effect as planned and the validity of such proposals will not be impacted by the tenure ending.
- Since different preset system parameters have different enabling conditions/methods, a proposals’ entry into force does not mean system parameters will be modified immediately.

9.6 Resignation of Committee Members

- Candidates for committee/formal committee members can terminate their capacity as committee members and the system will delay the return of their deposit, which is currently delayed by one day.
- After a formal committee member resigns, their capacity within that tenure will not be impacted, but in the next tenure that committee member will lose all voting rights.
- After resignation, all votes will undergo zero clearing (coming into effect after the re-elections)

- If an application to become a committee member is filed during the delayed release period, the tokens not released will be locked again. However, cleared votes cannot be recovered.

Committee Related Operations

The following related operation commands can be executed in the interactive command line in `yoyow_client`

10.1 Creating Committee Account

```
# create_committee_member <owner_account> <pledge_amount> <pledge_asset_symbol> <url>  
↔<broadcast>  
create_committee_member 223331844 1000 YOYO "http://www.yoyow.org" true
```

For details, please refer to: `create_committee_member`

10.2 Updating Committee Account

```
# update_committee_member <committee_member_account> <pledge_amount> <pledge_asset_  
↔symbol> <url> <broadcast>  
update_committee_member 223331844 1000 YOYO "http://demo.yoyow.org" true
```

For details, please refer to: `update_committee_member`

10.3 Voting for Committee

```
# update_committee_member_votes <voting_account> <committee_members_to_add>  
↔<committee_members_to_remove> <broadcast>  
update_committee_member_votes 223331844 ["init1"], [], true
```

For details, please refer to: `update_committee_member_votes`

10.4 Getting Info of Committee Members

```
# get_committee_member <owner_account>  
get_committee_member 25997
```

For details, please refer to: `get_committee_member`

10.5 Checking History of Current Candidates

```
# list_committee_members <lower_bound> <limit> <order_by>  
list_committee_members 0 5 1
```

For details, please refer to: `list_committee_members`

10.6 Checking the Current Committee Proposals Not Implemented

```
list_committee_proposals
```

For details, please refer to: `list_committee_members`

YOYOW uses the DPoS consensus algorithm. All token holders are able to choose witnesses and the committee through the voting system, allowing them to make decisions about the YOYOW system.

11.1 Voting Qualification

- Voting qualification is assigned to those holders owning a certain amount of tokens. The current value is 10,000 YOYO. Please refer to the field `min_governance_voting_balance` in [Parameter Table](#).
- Accounts included in the committee's blacklist are disqualified from voting.

11.2 Validity Period of Voting

- Users qualified for voting will activate their intention to vote when they vote for the first time. The validity period is T , which is initially set at 90 days.
- If there is any new voting action within the validity period, it will be extended by T , which means, the expiry date will be changed to: the date of the updated voting action + T .
- If there is no new voting action within the validity period, it is assumed that there is no more voting intention.
- If users are disqualified from voting, or have no more voting intention, their votes will become void.

11.3 Voting Method

- Committee elections: one vote for one candidate—each account can only support one candidate at a time, and the voter can change the vote at any time
- Witness elections: one vote for numerous candidates—to strengthen consensus, each account can vote for numerous witnesses.

11.4 Valid Vote Counting

To enhance the fairness of voting, vote counting also takes into consideration the voters' number of tokens held, average token volume, time held and other factors.

- When the account activates the voting intention, it begins to accumulate the age of the tokens.
- The number of valid votes is equal to average balance over 60 days, or the current balance of the account (whichever is smaller), which means: number of valid votes = min (average balance within 60 days, current balance). That is to say, when the tokens are transferred in, the age of tokens gradually accumulates and increases the corresponding portion; when the tokens are transferred out, the age of tokens immediately decreases the corresponding portion.
- The number of votes for a candidate is equal to the total number of valid votes from supporters.
- As it takes a lot of effort to count based on the age of the tokens in all accounts, which are updated in real-time, a delayed updating mode is used for vote counting, which means there might be a few deviations from counting based on real-time statistics.

11.5 Voting Proxies

- A proxy can be designated for voting activities. For example: when Account A designates Account B as voting proxy, the voter of Account B will receive the valid vote(s) from both A and B. In this case, A is the client and B is the proxy.
- Multi-layered proxies are allowed. For example: Account A designates Account B as its voting proxy, and Account B designates Account C as its voting proxy. However, there are restrictions on the total number of proxy layers.
- Any account cannot designate itself as a voting proxy.
- Whenever there is an action of proxy voting, the validity period of the client's intention to vote will be updated.
- When the proxy has no intention to vote or no qualification to vote, the proxy relationship will be automatically terminated, and the client's voting status will be changed to "with no vote for anyone" while the validity period of the client's intention to vote remains the same.
- When a proxy reactivates their intention to vote, the proxy relationship is not recovered.

Currently, the collateral can be used for updating the platform accounts or applying for becoming miner witnesses. Only YOYO is accepted for the collateral.

12.1 Platform Collateral

Currently, the minimum collateral value for the platform is 10000 YOYO. The qualification of the platform is evaluated according to the current collateral value, which can be modified by the committee. Please refer to the parameter `platform_min_pledge` in [parameter table](#).

12.2 Committee Collateral

The minimum collateral value for the committee is 1000 YOYO.

12.3 Miner Witness Collateral

- The miner witness calculates the number of blocks to be produced according to the amount of valid collateral of each account. For example, if A, B and C all deposit collateral, and A has six tokens of valid collateral, B has three tokens of valid collateral, and C has one token of valid collateral, then, for the following 100 blocks, it should be arranged that A produces 60 blocks, B produces 30 blocks and C produces 10 blocks.
- The valid collateral will be the average collateral over seven days, or the current collateral (whichever is smaller). That is to say, when the collateral is increased, the valid collateral will increase gradually, and when the collateral decreases or is canceled, the valid collateral will immediately decrease.

12.4 Collateral Refunding

The refunding of collateral deposits will be delayed for a certain period of time when the collateral amount is decreased or the collateral is cancelled. Current value: 7 days.

Notes:

1. When there are continuous decreases in collateral, the amount will be accumulated and refunded in one lump sum. The time of refund will be delayed on the basis of the time of the last decrease.
2. For collateral increases, if there is a collateral deposit to be returned to the current account, that refund will be deducted first, and then the remaining amount will be deducted from the balance of the account.

The core asset of YOYOW is YOYO.

13.1 Balance

We recommend storing most of your YOYO in the balance. The YOYO in the balance can accumulate Bonus Points, which can be used to pay fees.

13.2 Tipping

We recommend storing small amount of YOYO in the Tipping. For daily small payments, it is recommended to use Tipping instead of balance.

13.3 The Design of Balance and Tipping

Balance and Tipping are designed primarily to balance usability and security. The use of balance and tipping requires two different permissions. The active key controls the balance, and the tipping key (secondary key) controls the tipping.

The authority of tipping can be delegated to the platform to facilitate automated trading in the ecosystem created by the platform, without having to use your own private key for each time so as to reduce risks.

Therefore, the balance is designed to store large amounts of assets, and it is not recommended to deposit too many assets in tipping.

User-Issued Assets (UIA)

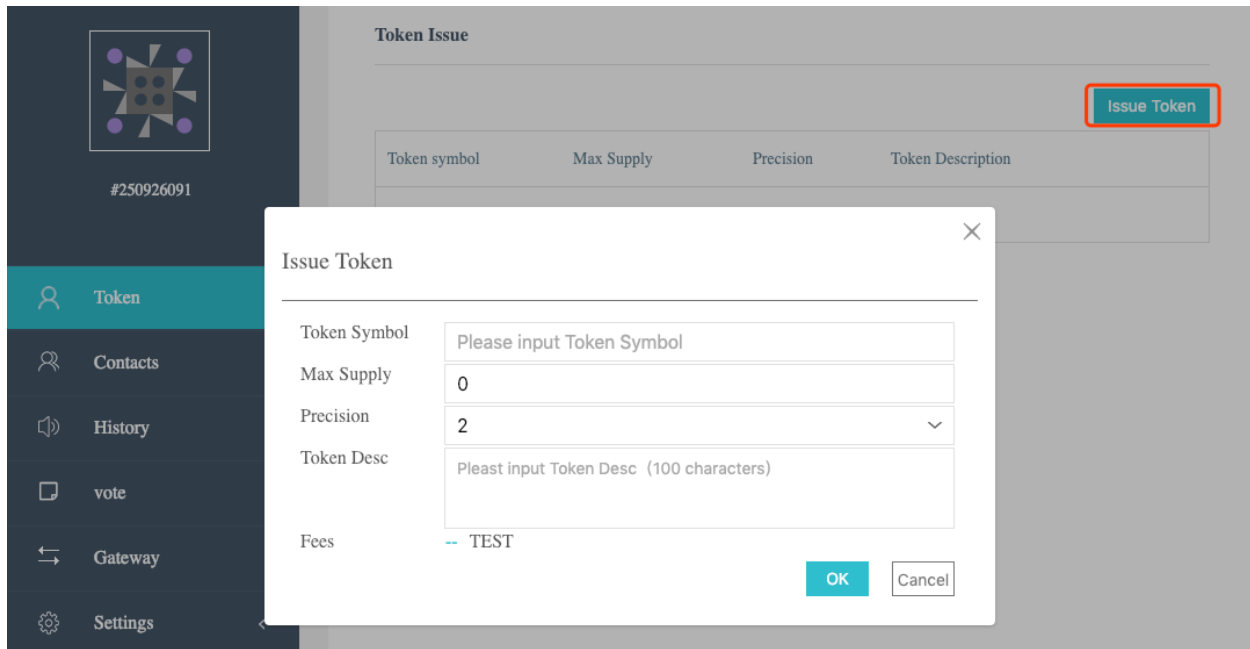
YOYOW provides the feature of issuing assets, and common accounts can create and issue a variety of custom assets (UIA).

User-issued assets can be transferred in transactions, so they can be used as rewards for content platforms to replace the traditional point mechanism.

Later, the exchange relationship between YOYO and UIA will be introduced.

14.1 Issuing Assets

The web wallet provides a simple operation for issuing assets, and it is convenient to specify the code for issuing the token, the total supply, and the number of decimal points. As shown below:



The fees for issuing assets include the basic fees and the cost per kilobyte. The basic fees are charged according to the length of the Token code. The current basic fees are as follows:

14.2 Advanced Options

In addition to the simple feature of UIA, users issuing assets can define more features.

- **Whitelist:** Assets can optionally be whitelisted. If a whitelist is defined, only whitelisted accounts can hold and use assets.
- **Blacklist:** The asset enables the blacklist by default. The account in the blacklist cannot hold and use the assets.
- **Forced transfers:** Assets can be set to allow for forced transfers. If enabled, the issuer of the asset can force transfer or take back such assets in other people's accounts.
- **Restricted transfers:** Assets can be set to allow for restricted transfers. If enabled, the originator or recipient of the transfer must be the issuer of the assets.
- **Issuance of assets:** After the asset is created, it needs to be issued to the account for circulation. If enabled, the issuer of the asset can add a certain amount of such assets to an account. The issue means increasing the current circulation of the asset.
- **Modifying the circulation limit:** If enabled, the asset issuer can modify the circulation limit of the assets.

For more information, please refer to the API documentation for a detailed description of the request parameters in the [Creation of Assets](#).

Bonus Points are rewards for YOYO holders. When your account holds YOYO, Bonus Points will be accumulated over time. There is an upper limit for Bonus Points accumulated, and you have to withdraw Bonus Points manually.

15.1 Using Bonus Points

At present, the only use of Bonus Points is for paying fees. You can use Bonus Points to offset fees. 100 Bonus Points can be used as 1 YOYO, under the circumstances.

15.2 Acquisition of Bonus Points

- When the “balance” of your account reaches a certain amount of YOYO, the system will automatically accumulate Bonus Points for you.
- Bonus Points are withdrawn manually. You can manually withdraw available Bonus Points to your account.
- Rate of Credit Accumulation: the current setting allows you to withdraw Bonus Points equivalent to 1 YOYO per day for every 10,000 YOYO.

15.3 Renting Bonus Points

A Bonus Point Rental function will be enabled in the future.

15.4 Gifting Bonus Points

Currently, when implementing the “withdrawing Bonus Points” action, you are able to select the account of a contact and withdraw the Bonus Points into their accounts.

Asset Related Operations

The following related operation commands can be executed in the interactive command line in `yoyow_client`

16.1 Common Transfer

```
# transfer <from> <to> <amount> <asset_symbol> <memo> <broadcast>
transfer 250926091 209414065 100 YOYO "memo" true
```

For details, please refer to: `transfer`

16.2 Query

16.2.1 Checking YOYO Balance in the Account

This function only returns the number of YOYO owned by the account.

```
# list_account_balances <account>
list_account_balances 250926091
```

For details, please refer to: `list_account_balances`

16.2.2 Checking All Assest Balances in the Account

This function does not have an interface for the wallet at the moment, and it can only call the node's api, such as using `wscat` to link the websocket port.

```
{"id":1, "method":"call", "params":[0,"get_account_balances",["250926091", []]]}
```

For details, please refer to: `get_account_balances`

16.3 About Bonus Points

16.3.1 Checking Bonus Points

Bonus Points can be viewed by using the command “get_full_account” and the “statistics” in the returned results include “csaf” “core balance” “prepaid”, corresponding to “Bonus Points” “Balance” “Tipping”.

```
# get_full_account <account_name_or_id>
get_full_account 250926091
```

16.3.2 Collecting the Bonus Points for the Time Being

```
# collect_csaf <from> <to> <amount> <asset_symbol> <broadcast>
collect_csaf 250926091 209414065 100 YOYO true
```

For details, please refer to: `collect_csaf`

16.3.3 Collecting the Bonus Points Accumulated before a Specific Time

```
# collect_csaf_with_time <from> <to> <amount> <asset_symbol> <time> <broadcast>
collect_csaf_with_time 250926091 209414065 100 YOYO "2018-04-16T02:44:00" true
```

For details, please refer to: `collect_csaf_with_time`

16.4 User-Issued Assets

There are two core operations for the user-issued assets, including creating assets (`create_asset`) and issuing assets (`issue_asset`). The newly created assets are not available (no circulation) and need to be issued to the specified account before they can be used.

16.4.1 Getting Assets Info

```
# get_asset <asset_name_or_id>
get_asset 3
```

For details, please refer to: `get_asset`

16.4.2 Checking Assets List

```
# list_assets <lower_bound_symbol> <limit>
list_assets YOYO 4
```

For details, please refer to: `list_assets`

16.4.3 Creating Assets

```
# create_asset <issuer> <symbol> <precision> <common> <initial_supply> <broadcast>
create_asset 250926091 TOTOTO 4 {"max_supply":300000,"market_fee_percent":0,"max_
↪market_fee":0,"issuer_permissions":4} 200000 true
```

For details, please refer to: `create_asset`

16.4.4 Issuing Assets

```
# issue_asset <to_account> <amount> <symbol> <memo> <broadcast>
issue_asset 250926091 10000 WOWO memo true
```

For details, please refer to: `issue_asset`

16.4.5 Updating Assets

```
# update_asset <symbol> <new_issuer> <new_options> <broadcast>
update_asset WOWO null {"max_supply":"20000000000"} true
```

For details, please refer to: `update_asset`

16.4.6 Destroying Assets

```
# reserve_asset <from> <amount> <symbol> <broadcast>
reserve_asset 250926091 1000 WOWO true
```

For details, please refer to: `reserve_asset`

Contents

- *Node API Descriptions*
 - *Node Connection Method*
 - *API Classification*
 - *Database API*
 - * *1.1.1 get_required_signatures*
 - * *1.1.2 get_accounts_by_uid*
 - * *1.1.3 get_account_balances*
 - * *1.1.4 get_post*
 - * *1.1.5 get_posts_by_platform_poster*
 - * *1.1.6 get_required_fee_data*
 - * *1.1.7 get_full_accounts_by_uid*
 - * *1.1.8 get_witness_by_account*
 - * *1.1.9 get_witnesses*
 - * *1.1.10 lookup_witnesses*
 - * *1.1.11 get_committee_member_by_account*
 - * *1.1.12 get_committee_members*
 - * *1.1.13 lookup_committee_members*
 - * *1.1.14 list_committee_proposals*
 - * *1.1.15 lookup_accounts_by_name*

- * 1.1.16 *get_platform_by_account*
 - * 1.1.17 *get_platforms*
 - * 1.1.18 *lookup_platforms*
 - * 1.1.19 *get_platform_count*
 - * 1.1.20 *get_assets*
 - * 1.1.21 *list_assets*
 - * 1.1.22 *lookup_asset_symbols*
- *History API*
- * 1.2.1 *get_relative_account_history*

17.1 Node Connection Method

Testing Environment

```
websocket interface address ws://47.52.155.181:10011
jsonrpc interface address http://47.52.155.181:10011/rpc
```

Formal Environment

```
websocket interface address ws://139.198.1.234:9000
jsonrpc interface address http://139.198.1.234:9000/rpc
```

use wscat to connect **wscat installation method** `wscat -c ws://47.52.155.181:10011`

use curl post data to connect `curl -data '{"jsonrpc": "2.0", "method": "call", "params": [0, "get_accounts_by_uid", [{"250926091"}]], "id": 1}' http://47.52.155.181:10011/rpc`

17.2 API Classification

YOYOW's API classification is similar to BTS. The following focuses on database api and history api. When requesting via websocket, the parameter is a json string in the following format:

```
{ "id": 1, "method": "call", "params": [API level, "function name", [specific parameters]] }
```

In use, you need to fill in API level, function name, and specific parameters, of which the API level can be obtained by sending the following strings via websocket:

```
{ "id": 2, "method": "call", "params": [1, "history", []] }
```

For example, the above request will return:

```
{
  "id": 2,
  "jsonrpc": "2.0",
  "result": 2
}
```

Here the result is 2, which means that when using the history api, the API level needs to be filled in 2. (Note: not every YOYOW node returns the same configuration; it depends on each node's limit on exposing the API)

The database API can be called directly by default by specifying the API level to 0 or by querying the result value through using the following strings:

```
{“id”:2, “method”:“call”, “params”:[1,“database”,[]]}
```

17.3 Database API

17.3.1 1.1.1 get_required_signatures

Return the 3 sets associated with signing the transaction based on the given transaction (which may already contain the signature), and the given set of spare public keys:

A subset of the available spare public key `set`, can be used to sign the transaction
 Public key that may also be needed (**not in** the signature, nor **in** the `set` of spare_
 ↪public keys)
 Extra signature already included **in** the transaction

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

trx transaction, may have already contained signature

available_keys array of public keys For example[“YYW5eDSFYeiqyFRajfPP8tTZMm7fUeyc7H65zmnHtDW4SQJdwqTBD”]

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0,"get_required_signatures", [{"operations":[[0, {
↪"fee":{"total":{"amount":100000,"asset_id":0}}, {"from":250926091, "to":223331844,
↪"amount":{"amount":100000,"asset_id":0}, {"extensions":{}}]]}], [
↪"YYW5eDSFYeiqyFRajfPP8tTZM7mfUeyc7H65zmnHtDW4SQJdwqTBD"]}]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "get_accounts_by_uid",
↪ [{"operations": [[0, {"fee": {"total": {"amount": 100000, "asset_id": 0}}, "from
↪ ": 250926091, "to": 223331844, "amount": {"amount": 100000, "asset_id": 0}, "extensions": {}
↪ ]]}], [{"YYW5eDSFYeiqyFRajfPP8tTZM7mfUeyc7H65zmnHtDW4SQJdwqTBD"}]]], "id": 1}' http://
↪ 47.52.155.181:10011/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    [
      [
        "YYW5eDSFYeiqyFRajfPP8tTZM7mfUeyc7H65zmnHtDW4SQJdwqTBD"
      ], //a subset of the available spare public key set, can be used to sign the
↪ transaction
      [
        "YYW6fU7Th8uESW9FZnpkhYaTUwtSvn3f2TQxFVA3ef2SSiwdZES71",
        "YYW7UoNSEQAUPvnvoBRVKyPAD9845esnpiK6MgHinsn5yqr5UgT5W"
      ] //public key that may also be needed (not in the signature, nor in the set of
↪ spare public keys)
    ],
    [] // extra signature already included in the transaction
  ]
}
```

17.3.2 1.1.2 get_accounts_by_uid

Return multiple account information based on uid. The quantity must be ≤ 1000 .

If the uid does not exist, the corresponding position result is null .

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

account_uids uid arraylength is less than 1000 for example["250926091"]

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0,"get_accounts_by_uid",["250926091"]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "get_accounts_by_uid",
↪ ["250926091"]], "id": 1}' http://47.52.155.181:10011/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    {
      "id": "1.2.1378",
      "uid": 250926091,
      "name": "yoyo250926091",
      "owner": {
        "weight_threshold": 1,
        "account_uid_auths": [],
        "key_auths": [
          ["YYW7UoNSEQAUPvnvoBRVKyPAD9845esnpiK6MgHinsn5yqr5UgT5W", 1]
        ]
      },
      "active": {
        "weight_threshold": 1,
        "account_uid_auths": [],
        "key_auths": [
          ["YYW6fU7Th8uESW9FZnpkhYaTUwtSvn3f2TQxFVA3ef2SSiwdZES71", 1]
        ]
      },
      "secondary": {
        "weight_threshold": 1,
        "account_uid_auths": [],
        "key_auths": [
          ["YYW5eDSFYeiqyFRajfPP8tTZM7mfUeyc7H65zmnHtDW4SQJdwqTBD", 1]
        ]
      },
      "memo_key": "YYW7SpC4QLY1LRRxFQ2hbYHdAyQo88L8qnPJcDJkiRMugcnFGUGvo",
      "reg_info": {
        "registrar": 206336051,
        "referrer": 25997,
```

(continues on next page)

(continued from previous page)

```
    "registrar_percent": 0,
    "referrer_percent": 0,
    "allowance_per_article":
    {
        "amount": 0,
        "asset_id": 0
    },
    "max_share_per_article":
    {
        "amount": 0,
        "asset_id": 0
    },
    "max_share_total":
    {
        "amount": 0,
        "asset_id": 0
    },
    "buyout_percent": 10000
},
"can_post": true,
"can_reply": false,
"can_rate": false,
"is_full_member": true,
"is_registrar": false,
"is_admin": false,
"create_time": "2018-04-03T08:21:00",
"last_update_time": "2018-04-03T08:21:00",
"active_data": "{}",
"secondary_data": "{}",
"statistics": "2.5.1378"
}}
}
```

17.3.3 1.1.3 get_account_balances

Query the asset balance based on uid and asset type.

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

uid uid for example: "250926091"

assets a list of asset type id, with 0 representing core assets. For example: [0,1]. If the value is empty ([]), return all asset balances in the account

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0,"get_account_balances",["250926091", [0,1]]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "get_account_balances", ["250926091", [0,1]]], "id": 1}' http://47.52.155.181:10011/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    {
      "amount": 1099970704,
      "asset_id": 0
    },
    {
      "amount": 0,
      "asset_id": 1
    }
  ]
}
```

17.3.4 1.1.4 get_post

Return post information based on platform owner uid, poster uid, and post pid.

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

platform_owner platform owner id

poster_uid poster id

post_pid post id (for exmaple1)

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0,"get_post",["223331844",223331844,0,1]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "get_post", [{"
↪ "250926091"}]]}, {"id": 1}]' http://47.52.155.181:10011/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "id": "1.7.14",
    "platform": 223331844,
    "poster": 223331844,
    "post_pid": 1,
    "hash_value": "asdfasdfasdfasdf",
    "extra_data": "{}",
    "title": "post a",
    "body": "post b",
    "create_time": "2018-05-03T12:40:39",
    "last_update_time": "2018-05-03T12:40:39"
  }
}
```

17.3.5 1.1.5 get_posts_by_platform_poster

Query the list of posts according to the platform owner uid, poster uid, post time period.

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

platform_owner platform owner id

poster_uid poster id; poster_uid can be null, query all users' posts at this time.

create_time_range The time limit is limited by two time points whose time orders are not limited. The query range is like this: the earliest time < Posting time <= latest time.

limit the number is limited, not exceeding 100.

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0,"get_posts_by_platform_poster",[223331844, null,
↪ ["2018-04-03T12:42:36","2018-05-03T12:42:36"], 100]}}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "get_accounts_by_uid",
↪ [223331844, null, ["2018-04-03T12:42:36","2018-05-03T12:42:36"], 100]], "id": 1}'
↪http://47.52.155.181:10011/rpc
```

Return results

The results are sorted by time, with the latest one being the top. If the time is the same, the results are sorted by the actual order of receiving blocks, with the later block reception being in the front.

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    {
```

(continues on next page)

(continued from previous page)

```

    "id": "1.7.14",
    "platform": 223331844,
    "poster": 223331844,
    "post_pid": 1,
    "hash_value": "asdfasdfasdf",
    "extra_data": "{}",
    "title": "post a",
    "body": "post b",
    "create_time": "2018-05-03T12:40:39",
    "last_update_time": "2018-05-03T12:40:39"
  }
]
}

```

17.3.6 1.1.6 get_required_fee_data

Give a set of operations, return the fee information required for the operation. The API only supports core assets.

wherein

```

required_fee_data
{
  account_uid_type fee_payer_uid; // payer uid
  int64_t          min_fee;      // The lowest total cost and the unit is the value
↳of core asset after being removed the part after the decimal point (same usage as
↳the asset type)
  int64_t          min_real_fee; // The lowest real cost (the part that cannot be
↳deducted using tokens), and the unit is the same as above
};

```

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

ops uid arraythe length is less than 1000; for example["250926091"]

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0, "get_required_fee_data", [[[0, {"fee":{"total":{"
↪"amount":200000,"asset_id":0},"options":{"from_balance":{"amount":200000,"asset_id
↪":0}}},"from":236542328,"to":228984329,"amount":{"amount":100000,"asset_id":0},
↪"extensions":{"from_balance":{"amount":100000,"asset_id":0},"to_balance":{"amount
↪":100000,"asset_id":0}}]]]]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "get_required_fee_data
↪", [[[0, {"fee":{"total":{"amount":200000,"asset_id":0},"options":{"from_balance":{"
↪"amount":200000,"asset_id":0}}},"from":236542328,"to":228984329,"amount":{"amount
↪":100000,"asset_id":0},"extensions":{"from_balance":{"amount":100000,"asset_id":0},
↪"to_balance":{"amount":100000,"asset_id":0}}]]]]], "id": 1}' http://47.52.155.
↪181:10011/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    {
      "fee_payer_uid": 236542328,
      "min_fee": 20000,
      "min_real_fee": 0
    }
  ]
}
```

17.3.7 1.1.7 get_full_accounts_by_uid

Get the corresponding information based on a set of account uid.

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

uids uid array the length is less than 1000; for example["250926091"]

options options array

Options The array can have the following parameters

```
{
optional fetch_account_object;
optional fetch_statistics;
optional fetch_csaf_leases_in;
optional fetch_csaf_leases_out;
optional fetch_voter_object;
optional fetch_witness_object;
optional fetch_witness_votes;
optional fetch_committee_member_object;
optional fetch_committee_member_votes;
optional fetch_platform_object;
optional fetch_platform_votes;
optional fetch_assets;
optional fetch_balances;
}
```

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0, "get_full_accounts_by_uid", [{"250926091"},{}
↪]}

{"id":1, "method":"call", "params":[0, "get_full_accounts_by_uid", [{"223331844"},{
↪"fetch_assets": true}]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "get_full_accounts_by_
↪uid", [{"250926091"},{}]], "id": 1}' http://47.52.155.181:10011/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    [250926091,
     {
       "account":
       {
```

(continues on next page)

(continued from previous page)

```
"id": "0.0.0",
"uid": 0,
"name": "",
"owner":
{
  "weight_threshold": 0,
  "account_uid_auths": [],
  "key_auths": []
},
"active":
{
  "weight_threshold": 0,
  "account_uid_auths": [],
  "key_auths": []
},
"secondary":
{
  "weight_threshold": 0,
  "account_uid_auths": [],
  "key_auths": []
},
"memo_key": "YYW1111111111111111111111111111111114T1Anm",
"reg_info":
{
  "registrar": 1264,
  "referrer": 1264,
  "registrar_percent": 0,
  "referrer_percent": 0,
  "allowance_per_article":
  {
    "amount": 0,
    "asset_id": 0
  },
  "max_share_per_article":
  {
    "amount": 0,
    "asset_id": 0
  },
  "max_share_total":
  {
    "amount": 0,
    "asset_id": 0
  },
  "buyout_percent": 10000
},
"can_post": true,
"can_reply": false,
"can_rate": false,
"is_full_member": false,
"is_registrar": false,
"is_admin": false,
"create_time": "1970-01-01T00:00:00",
"last_update_time": "1970-01-01T00:00:00",
"active_data": "{}",
"secondary_data": "{}",
"statistics": "2.5.0"
},
```

(continues on next page)

(continued from previous page)

```

    "statistics":
    {
      "id": "0.0.0",
      "owner": 31120496,
      "total_ops": 0,
      "removed_ops": 0,
      "prepaid": 0,
      "csaf": 0,
      "core_balance": 0,
      "core_leased_in": 0,
      "core_leased_out": 0,
      "average_coins": 0,
      "average_coins_last_update": "1970-01-01T00:00:00",
      "coin_seconds_earned": "0",
      "coin_seconds_earned_last_update": "1970-01-01T00:00:00",
      "total_witness_pledge": 0,
      "releasing_witness_pledge": 0,
      "witness_pledge_release_block_number": 4294967295,
      "last_witness_sequence": 0,
      "uncollected_witness_pay": 0,
      "witness_last_confirmed_block_num": 0,
      "witness_last_aslot": 0,
      "witness_total_produced": 0,
      "witness_total_missed": 0,
      "witness_last_reported_block_num": 0,
      "witness_total_reported": 0,
      "total_committee_member_pledge": 0,
      "releasing_committee_member_pledge": 0,
      "committee_member_pledge_release_block_number": 4294967295,
      "last_committee_member_sequence": 0,
      "can_vote": true,
      "is_voter": false,
      "last_voter_sequence": 0,
      "last_platform_sequence": 0,
      "total_platform_pledge": 0,
      "releasing_platform_pledge": 0,
      "platform_pledge_release_block_number": 4294967295,
      "last_post_sequence": 0
    },
    "csaf_leases_in": [],
    "csaf_leases_out": [],
    "witness_votes": [],
    "committee_member_votes": []
  }
}

```

Return field description

return the structure definition of `full_account` in map as

```

full_account
{
  account;           // account basic info
  statistics;       // account dynamic info
}

```

(continues on next page)

(continued from previous page)

```

csaf_leases_in;           // fee token age borrowing details
csaf_leases_out;         // fee token age lending details
voter;                   // summary of account voting information
witness;                 // witness information
witness_votes;          // witness vote details (voting votes)
committee_member;       // committee candidate info
committee_member_votes; // committee election voting details (voting votes)
platform;               // platform information owned by this account
platform_votes;         // platform voting details (voting votes)
assets;                 // this account is the asset issuer's asset type id list
balances;               // balance sheet
};

```

17.3.8 1.1.8 get_witness_by_account

Give the uid of an account, return the corresponding witness information

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

account uid array the length is less than 1000; for example["250926091"]

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0,"get_witness_by_account",["132826789"]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "get_witness_by_
↪account", ["132826789"], "id": 1]}' http://47.52.155.181:10011/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result":
  {
    "id": "1.5.31",
    "account": 132826789,
    "name": "yoyo132826789",
    "sequence": 1,
    "is_valid": true,
    "signing_key": "YYW1111111111111111111111111111111114T1Anm",
    "pledge": "7500000000",
    "pledge_last_update": "2017-09-05T11:39:03",
    "average_pledge": "7500000000",
    "average_pledge_last_update": "2017-09-06T12:05:36",
    "average_pledge_next_update_block": 4294967295,
    "total_votes": 719683655,
    "by_pledge_position": "0",
    "by_pledge_position_last_update": "0",
    "by_pledge_scheduled_time": "45370982250075664161773192435",
    "by_vote_position": "0",
    "by_vote_position_last_update": "0",
    "by_vote_scheduled_time": "472822140789228182032488184547",
    "last_confirmed_block_num": 8168,
    "last_aslot": 8599,
    "total_produced": 25,
    "total_missed": 0,
    "url": ""
  }
}
```

Return field descriptions

The corresponding field data is only included in the returned result if the corresponding option in options is true. Among them, the token age borrowing details and lending details are only returned for the top 100 items.

If uid does not exist, there is no corresponding uid in the returned map.

17.3.9 1.1.9 get_witnesses

Give a set of uids, return the corresponding witness information

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

account_uids uid arrayfor example[132826789,25997]

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0, "get_witnesses", [[132826789,25997]]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "get_witnesses", [
↪ [[132826789,25997]]], "id": 1}' http://47.52.155.181:10011/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    {
      "id": "1.5.31",
      "account": 132826789,
      "name": "yoyo132826789",
      "sequence": 1,
      "is_valid": true,
      "signing_key": "YYW1111111111111111111111111111111114T1Anm",
      "pledge": "7500000000",
      "pledge_last_update": "2017-09-05T11:39:03",
      "average_pledge": "7500000000",
      "average_pledge_last_update": "2017-09-06T12:05:36",
      "average_pledge_next_update_block": 4294967295,
      "total_votes": 719683655,
      "by_pledge_position": "0",
      "by_pledge_position_last_update": "0",
      "by_pledge_scheduled_time": "45370982250075664161773192435",
      "by_vote_position": "0",
      "by_vote_position_last_update": "0",
      "by_vote_scheduled_time": "472822140789228182032488184547",
      "last_confirmed_block_num": 8168,
      "last_aslot": 8599,
      "total_produced": 25,
```

(continues on next page)

(continued from previous page)

```
    "total_missed": 0,
    "url": ""
  },
  {
    "id": "1.5.1",
    "account": 25997,
    "name": "init1",
    "sequence": 1,
    "is_valid": true,
    "signing_key": "YyW71suPihtG7jJAGiVBCkd63ECHYebQaPa894oy3r54zk3eMlitt",
    "pledge": 1000000000,
    "pledge_last_update": "2017-09-12T21:02:45",
    "average_pledge": 1000000000,
    "average_pledge_last_update": "2017-09-13T21:20:30",
    "average_pledge_next_update_block": 4294967295,
    "total_votes": 0,
    "by_pledge_position": "0",
    "by_pledge_position_last_update": "0",
    "by_pledge_scheduled_time": "340282366580656096882718510549",
    "by_vote_position": "0",
    "by_vote_position_last_update": "0",
    "by_vote_scheduled_time": "340282366920938463463374607431768211455",
    "last_confirmed_block_num": 5937330,
    "last_aslot": 6308879,
    "total_produced": 513249,
    "total_missed": 32165,
    "url": ""
  }
}]
}
```

17.3.10 1.1.10 lookup_witnesses

List current valid witnesses

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

lower_bound_uid Start the query with this as the starting uid, set it to 0 and start from the beginning.

limit Return quantity limit, up to 101

ops Sort type; value range [0, 1, 2]. 0:Sort by uid from big to small; 1: Sort by number of votes; 2: Sort by collateral amount

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0, "lookup_witnesses", [0,2,1]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "lookup_witnesses", ↵
↳[0,2,1]], "id": 1}' http://47.52.155.181:10011/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    {
      "id": "1.5.31",
      "account": 132826789,
      "name": "yoyo132826789",
      "sequence": 1,
      "is_valid": true,
      "signing_key": "YYW111111111111111111111111111111114T1Anm",
      "pledge": "7500000000",
      "pledge_last_update": "2017-09-05T11:39:03",
      "average_pledge": "7500000000",
      "average_pledge_last_update": "2017-09-06T12:05:36",
      "average_pledge_next_update_block": 4294967295,
      "total_votes": 701297305,
      "by_pledge_position": "0",
      "by_pledge_position_last_update": "0",
      "by_pledge_scheduled_time": "45370982250075664161773192435",
      "by_vote_position": "0",
      "by_vote_position_last_update": "0",
      "by_vote_scheduled_time": "485218414514968154552378399456",
      "last_confirmed_block_num": 8168,
      "last_aslot": 8599,
      "total_produced": 25,
      "total_missed": 0,
      "url": ""
    },
    {
      "id": "1.5.1",
      "account": 25997,
      "name": "init1",

```

(continues on next page)

(continued from previous page)

```
    "sequence": 1,
    "is_valid": true,
    "signing_key": "YYW71suPihtG7jJAGiVBCKd63ECHYebQaPa894oy3r54zk3eMlitt",
    "pledge": 1000000000,
    "pledge_last_update": "2017-09-12T21:02:45",
    "average_pledge": 1000000000,
    "average_pledge_last_update": "2017-09-13T21:20:30",
    "average_pledge_next_update_block": 4294967295,
    "total_votes": 0,
    "by_pledge_position": "0",
    "by_pledge_position_last_update": "0",
    "by_pledge_scheduled_time": "340282366580656096882718510549",
    "by_vote_position": "0",
    "by_vote_position_last_update": "0",
    "by_vote_scheduled_time": "340282366920938463463374607431768211455",
    "last_confirmed_block_num": 5935462,
    "last_aslot": 6307011,
    "total_produced": 513079,
    "total_missed": 32165,
    "url": ""
  }
}
```

17.3.11 1.1.11 get_committee_member_by_account

Give a uid, return the corresponding committee candidate information

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

account uid; for example "250926091"

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0, "get_committee_member_by_account", [25997]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "get_committee_member_
↪by_account", [25997], "id": 1]}' http://47.52.155.181:10011/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result":
  {
    "id": "1.4.0",
    "account": 25997,
    "name": "init1",
    "sequence": 1,
    "is_valid": true,
    "pledge": 0,
    "total_votes": 0,
    "url": ""
  }
}
```

17.3.12 1.1.12 get_committee_members

Get the corresponding information based on a set of account uid.

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

committee_member_uids uid array; for example[25997,26264]

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0, "get_committee_members", [[25997,26264]]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "get_committee_members", [[25997,26264]]], "id": 1}' http://47.52.155.181:10011/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    {
      "id": "1.4.0",
      "account": 25997,
      "name": "init1",
      "sequence": 1,
      "is_valid": true,
      "pledge": 0,
      "total_votes": 0,
      "url": ""
    },
    {
      "id": "1.4.1",
      "account": 26264,
      "name": "init2",
      "sequence": 1,
      "is_valid": true,
      "pledge": 0,
      "total_votes": 0,
      "url": ""
    }
  ]
}
```

17.3.13 1.1.13 lookup_committee_members

List the current valid committee candidate list

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

lower_bound_uid Start the query with this as the starting uid, set it to 0 and start from the beginning

limit Return quantity limit, up to 101

ops Sort type, value range [0,1,2]. 0:Sort by uid from big to small; 1: Sort by number of votes; 2: Sort by collateral amount

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0, "lookup_committee_members", [0,2,1]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "lookup_committee_
↵members", [0,2,1]], "id": 1}'
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    {
      "id": "1.4.0",
      "account": 25997,
      "name": "init1",
      "sequence": 1,
      "is_valid": true,
      "pledge": 0,
      "total_votes": 0,
      "url": ""
    },
    {
      "id": "1.4.1",
      "account": 26264,
```

(continues on next page)

(continued from previous page)

```
    "name": "init2",
    "sequence": 1,
    "is_valid": true,
    "pledge": 0,
    "total_votes": 0,
    "url": ""
  }]
}
```

17.3.14 1.1.14 list_committee_proposals

List all the committee proposals that have not been successfully implemented, including those that are being voted on, have been voted through but have not yet reached the execution time.

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

null

Precautions

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0, "list_committee_proposals", []]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "list_committee_
↪ proposals", []], "id": 1}' http://47.52.155.181:10011/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": []
}
```

17.3.15 1.1.15 lookup_accounts_by_name

Find the account UID by name.

The normal account name is currently yoyo+uid

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

lower_bound_name Start the query with this as the starting name, set it to an empty string and start from the beginning.

limit Return quantity limit, up to 1001

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0, "lookup_accounts_by_", ["",2]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "lookup_accounts_by_
↵name", ["",2]], "id": 1}' http://47.52.155.181:10011/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    ["abit", 209414065],
    ["agaoye", 209415129]
  ]
}
```

17.3.16 1.1.16 get_platform_by_account

Give a uid, return the platform information owned by the corresponding account

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access and authorization limit

null

Request parameters

account one account uid

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0, "get_platform_by_account", [224006453]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "get_platform_by_
↵account", [224006453]], "id": 1}' http://47.52.155.181:10011/rpc
```


Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "id": "1.6.4",
    "owner": 224006453,
    "name": "dwgMarket",
    "sequence": 1,
    "is_valid": true,
    "total_votes": 0,
    "url": "www.cad1688.com",
    "pledge": 1000000000,
    "pledge_last_update": "2018-04-04T08:38:24",
    "average_pledge": 0,
    "average_pledge_last_update": "2018-04-04T08:38:24",
    "average_pledge_next_update_block": 5712088,
    "extra_data": "{}",
    "create_time": "2018-04-04T08:38:24",
    "last_update_time": "1970-01-01T00:00:00"
  }
}
```

17.3.17 1.1.17 get_platforms

Give a set of uids, return the corresponding platform information; uid is the owner id of the platform

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

account_uids uid list [224006453,217895094]

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0, "get_platforms", [[224006453,217895094]]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "get_platforms", [
→[[224006453,217895094]]], "id": 1}' http://47.52.155.181:10011/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    {
      "id": "1.6.4",
      "owner": 224006453,
      "name": "dwgMarket",
      "sequence": 1,
      "is_valid": true,
      "total_votes": 0,
      "url": "www.cad1688.com",
      "pledge": 1000000000,
      "pledge_last_update": "2018-04-04T08:38:24",
      "average_pledge": 0,
      "average_pledge_last_update": "2018-04-04T08:38:24",
      "average_pledge_next_update_block": 5712088,
      "extra_data": "{}",
      "create_time": "2018-04-04T08:38:24",
      "last_update_time": "1970-01-01T00:00:00"
    },
    {
      "id": "1.6.0",
      "owner": 217895094,
      "name": "test-yoyow",
      "sequence": 1,
      "is_valid": true,
      "total_votes": 0,
      "url": "http://demo.yoyow.org/",
      "pledge": 1000000000,
      "pledge_last_update": "2018-02-10T01:03:57",
      "average_pledge": 176601774,
      "average_pledge_last_update": "2018-02-11T06:49:12",
      "average_pledge_next_update_block": 4562164,
      "extra_data": "{\"login\":\"http://192.168.1.184:8280/login\"}",
      "create_time": "2018-02-10T01:03:57",
      "last_update_time": "2018-02-11T06:49:12"
    }
  ]
}
```

17.3.18 1.1.18 lookup_platforms

Query by platform owner to list the current valid platforms

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

lower_bound_uid Start the query with this as the starting uid, set it to 0 and start from the beginning.

limit Return quantity limit, up to 101

ops Sort type, value range [0,1,2], 0: Sort by uid from big to small; 1: Sort by number of votes; 2: Sort by collateral amount

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0, "lookup_platforms", [0,2,1]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "lookup_platforms", ↵
↵[0,2,1]], "id": 1}' http://47.52.155.181:10011/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    {
      "id": "1.6.0",
      "owner": 217895094,
```

(continues on next page)

(continued from previous page)

```
    "name": "test-yoyow",
    "sequence": 1,
    "is_valid": true,
    "total_votes": 0,
    "url": "http://demo.yoyow.org/",
    "pledge": 1000000000,
    "pledge_last_update": "2018-02-10T01:03:57",
    "average_pledge": 176601774,
    "average_pledge_last_update": "2018-02-11T06:49:12",
    "average_pledge_next_update_block": 4562164,
    "extra_data": "{\"login\": \"http://192.168.1.184:8280/login\"}",
    "create_time": "2018-02-10T01:03:57",
    "last_update_time": "2018-02-11T06:49:12"
  },
  {
    "id": "1.6.4",
    "owner": 224006453,
    "name": "dwgMarket",
    "sequence": 1,
    "is_valid": true,
    "total_votes": 0,
    "url": "www.cad1688.com",
    "pledge": 1000000000,
    "pledge_last_update": "2018-04-04T08:38:24",
    "average_pledge": 0,
    "average_pledge_last_update": "2018-04-04T08:38:24",
    "average_pledge_next_update_block": 5712088,
    "extra_data": "{}",
    "create_time": "2018-04-04T08:38:24",
    "last_update_time": "1970-01-01T00:00:00"
  }
}]
}
```

17.3.19 1.1.19 get_platform_count

Return the total number of platforms

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

null

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0, "get_platform_count", []]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "get_platform_count", ↵
↵[]], "id": 1}' http://47.52.155.181:10011/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": 5
}
```

17.3.20 1.1.20 get_assets

Give a set of asset ids, return the details of the corresponding assets.

Parameters `asset_ids` a set of assets id

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

`asset_ids` asset id array; for the time being, only the core asset YOYO is accepted for example [0]

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0, "get_assets", [[0]]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "get_assets", [[0]]],
↵"id": 1}' http://47.52.155.181:10011/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    {
      "id": "1.3.0",
      "asset_id": 0,
      "symbol": "YOYO",
      "precision": 5,
      "issuer": 1264,
      "options": {
        "max_supply": "2000000000000000",
        "market_fee_percent": 0,
        "max_market_fee": "1000000000000000",
        "issuer_permissions": 0,
        "flags": 0,
        "whitelist_authorities": [],
        "blacklist_authorities": [],
        "whitelist_markets": [],
        "blacklist_markets": [],
        "description": ""
      },
      "dynamic_asset_data_id": "2.2.0",
      "dynamic_asset_data": {
        "id": "2.2.0",
        "asset_id": 0,
        "current_supply": "106899730634997",
        "accumulated_fees": 0
      }
    }
  ]
}
```

The `dynamic_asset_data` field **in** the returned result includes the asset dynamic data_↵
↵details.

17.3.21 1.1.21 list_assets

Query asset details by page. The returned results are sorted in ASCII code order of the asset code.

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

lower_bound_symbol Start with this as the starting code

limit Return quantity limit, up to 101

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0, "list_assets", ["YOY",4]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "list_assets", ["YOY",
↪4]], "id": 1}' http://47.52.155.181:10011/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    {
      "id": "1.3.91",
      "asset_id": 91,
      "symbol": "YOYES",
      "precision": 2,
```

(continues on next page)

(continued from previous page)

```
"issuer": 215074501,
"options": {
  "max_supply": 1200,
  "market_fee_percent": 0,
  "max_market_fee": 1200,
  "issuer_permissions": 79,
  "flags": 0,
  "whitelist_authorities": [],
  "blacklist_authorities": [],
  "whitelist_markets": [],
  "blacklist_markets": [],
  "description": ""
},
"dynamic_asset_data_id": "2.2.91",
"dynamic_asset_data": {
  "id": "2.2.91",
  "asset_id": 91,
  "current_supply": 0,
  "accumulated_fees": 0
}
},
{
  "id": "1.3.130",
  "asset_id": 130,
  "symbol": "YOYIO",
  "precision": 2,
  "issuer": 254208024,
  "options": {
    "max_supply": 1258000000,
    "market_fee_percent": 0,
    "max_market_fee": 1258000000,
    "issuer_permissions": 79,
    "flags": 0,
    "whitelist_authorities": [],
    "blacklist_authorities": [],
    "whitelist_markets": [],
    "blacklist_markets": [],
    "description": ""
  },
  "dynamic_asset_data_id": "2.2.130",
  "dynamic_asset_data": {
    "id": "2.2.130",
    "asset_id": 130,
    "current_supply": 1258000000,
    "accumulated_fees": 0
  }
},
{
  "id": "1.3.0",
  "asset_id": 0,
  "symbol": "YOYO",
  "precision": 5,
  "issuer": 1264,
  "options": {
    "max_supply": "2000000000000000",
    "market_fee_percent": 0,
    "max_market_fee": "1000000000000000",
```

(continues on next page)

(continued from previous page)

```

    "issuer_permissions": 0,
    "flags": 0,
    "whitelist_authorities": [],
    "blacklist_authorities": [],
    "whitelist_markets": [],
    "blacklist_markets": [],
    "description": ""
  },
  "dynamic_asset_data_id": "2.2.0",
  "dynamic_asset_data": {
    "id": "2.2.0",
    "asset_id": 0,
    "current_supply": "106899950291573",
    "accumulated_fees": 0
  }
},
{
  "id": "1.3.2",
  "asset_id": 2,
  "symbol": "YOYOW",
  "precision": 5,
  "issuer": 25638,
  "options": {
    "max_supply": "1000000000000",
    "market_fee_percent": 0,
    "max_market_fee": "1000000000000",
    "issuer_permissions": 79,
    "flags": 0,
    "whitelist_authorities": [],
    "blacklist_authorities": [],
    "whitelist_markets": [],
    "blacklist_markets": [],
    "description": ""
  },
  "dynamic_asset_data_id": "2.2.2",
  "dynamic_asset_data": {
    "id": "2.2.2",
    "asset_id": 2,
    "current_supply": 0,
    "accumulated_fees": 0
  }
}
]
}

```

17.3.22 1.1.22 lookup_asset_symbols

Give a set of asset codes or ids, return the details of the corresponding assets.

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

symbols_or_ids Array form, the symbol code or ID of the asset to be retrieved, for example: ["YOYO"]
or [0]

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":[0, "lookup_asset_symbols", [[0]]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "lookup_asset_symbols
↵", [[0]]], "id": 1}' http://47.52.155.181:10011/rpc
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "lookup_asset_symbols
↵", [{"YOYO"}]], "id": 1}' http://47.52.155.181:10011/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    {
      "id": "1.3.0",
      "asset_id": 0,
      "symbol": "YOYO",
      "precision": 5,
      "issuer": 1264,
      "options": {
        "max_supply": "2000000000000000",
        "market_fee_percent": 0,
        "max_market_fee": "1000000000000000",
        "issuer_permissions": 0,
        "flags": 0,
        "whitelist_authorities": [],
        "blacklist_authorities": [],
```

(continues on next page)

(continued from previous page)

```
    "whitelist_markets": [],
    "blacklist_markets": [],
    "description": ""
  },
  "dynamic_asset_data_id": "2.2.0",
  "dynamic_asset_data": {
    "id": "2.2.0",
    "asset_id": 0,
    "current_supply": "106900048605605",
    "accumulated_fees": 0
  }
}
]
```

17.4 History API

17.4.1 1.2.1 get_relative_account_history

Get account history

Supported format

JSON

Request method

WebSocket; JSON-RPC

Access authorization limit

null

Request parameters

account can be uid or account nickname

op_type the type of limited operation, see the type of operation. When the value is null, all operation types are returned; when 0, all transfer operations are available.

start Query start numbersequence number

limit Return the total number of results

end When the value is 0, the most recent history of operations can be obtained.

The number of returned results will be in the end - start range; if the limit value is smaller than end - start, the latest operation record that satisfies the limit condition is returned. The returned results are sorted in the way that the latest ones are returned first.

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://47.52.155.181:10011
{"id":1, "method":"call", "params":["history", "get_relative_account_history",
↪[223331844, null, 1,3,10]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": ["history", "get_relative_
↪account_history", [223331844, null, 1,3,10]], "id": 1}' http://47.52.155.181:10011/
↪rpc
```

Return results

Each piece of data in the returned list is a pair type. The first element in the pair is the sequence number recorded in the account history, and the second element is the specific operation.

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    [
      10,
      {
        "id": "1.12.48157",
        "op": [
          0,
          {
            "fee": {
              "total": {
                "amount": 20000,
                "asset_id": 0
              },
            "options": {
              "from_csaf": {
                "amount": 20000,
                "asset_id": 0
              }
            }
          },
        ],
        "from": 217895094,
        "to": 223331844,
        "amount": {
          "amount": "200000000000",
          "asset_id": 0
        },
        "extensions": {
          "from_balance": {
            "amount": "200000000000",
```

(continues on next page)

(continued from previous page)

```

        "asset_id": 0
      },
      "to_balance": {
        "amount": "200000000000",
        "asset_id": 0
      }
    }
  ],
  "result": [
    0,
    {}
  ],
  "block_timestamp": "2018-05-02T09:24:30",
  "block_num": 6515279,
  "trx_in_block": 0,
  "op_in_trx": 0,
  "virtual_op": 2715
}
],
[
  9,
  {
    "id": "1.12.47189",
    "op": [
      22,
      {
        "fee": {
          "total": {
            "amount": 200000,
            "asset_id": 0
          },
          "options": {
            "from_csaf": {
              "amount": 200000,
              "asset_id": 0
            }
          }
        },
        "voter": 236542328,
        "platform_to_add": [
          223331844
        ],
        "platform_to_remove": []
      }
    ],
    "result": [
      0,
      {}
    ],
    "block_timestamp": "2018-04-16T08:14:57",
    "block_num": 6053313,
    "trx_in_block": 0,
    "op_in_trx": 0,
    "virtual_op": 1157
  }
],
],

```

(continues on next page)

(continued from previous page)

```
[
  8,
  {
    "id": "1.12.47149",
    "op": [
      22,
      {
        "fee": {
          "total": {
            "amount": 200000,
            "asset_id": 0
          },
          "options": {
            "from_csaf": {
              "amount": 200000,
              "asset_id": 0
            }
          }
        },
        "voter": 250926091,
        "platform_to_add": [
          223331844
        ],
        "platform_to_remove": []
      }
    ],
    "result": [
      0,
      {}
    ],
    "block_timestamp": "2018-04-16T03:19:36",
    "block_num": 6049807,
    "trx_in_block": 0,
    "op_in_trx": 0,
    "virtual_op": 1117
  }
]
}
```

Contents

- *Wallet API Descriptions*
 - *Wallet Connection Method*
 - * *Download Method*
 - * *Starting Method*
 - *2.1 Tools API*
 - * *2.1.1 calculate_account_uid*
 - * *2.1.2 suggest_brain_key*
 - * *2.1.3 get_transaction_id*
 - *2.2 Queries API*
 - * *2.2.1 get_account*
 - * *2.2.2 get_full_account*
 - * *2.2.3 get_relative_account_history*
 - * *2.2.4 list_account_balances*
 - * *2.2.5 list_accounts_by_name*
 - * *2.2.6 get_witness*
 - * *2.2.7 list_witnesses*
 - * *2.2.8 get_committee_member*
 - * *2.2.9 list_committee_members*
 - * *2.2.10 list_committee_proposals*

- * 2.2.11 *get_platform_count*
- * 2.2.12 *get_platform*
- * 2.2.13 *list_platforms*
- * 2.2.14 *get_asset*
- * 2.2.15 *list_assets*
- 2.3 *Wallet/Private Key Management API*
 - * 2.3.1 *save_wallet_file*
 - * 2.3.2 *set_password*
 - * 2.3.3 *unlock*
 - * 2.3.4 *lock*
 - * 2.3.5 *import_key*
 - * 2.3.6 *dump_private_keys*
 - * 2.3.7 *list_my_accounts_cached*
- 2.4 *Operations/Transactions API*
 - * 2.4.1 *transfer*
 - * 2.4.2 *create_witness*
 - * 2.4.3 *update_witness*
 - * 2.4.4 *create_committee_member*
 - * 2.4.5 *update_committee_member*
 - * 2.4.6 *set_voting_proxy*
 - * 2.4.7 *update_witness_votes*
 - * 2.4.8 *update_committee_member_votes*
 - * 2.4.9 *collect_csaf_with_time*
 - * 2.4.10 *collect_csaf*
 - * 2.4.11 *create_platform*
 - * 2.4.12 *update_platform*
 - * 2.4.13 *update_platform_votes*
 - * 2.4.14 *account_auth_platform*
 - * 2.4.15 *account_cancel_auth_platform*
 - * 2.4.16 *create_asset*
 - * 2.4.17 *update_asset*
 - * 2.4.18 *enable_allowed_assets*
 - * 2.4.19 *update_allowed_assets*
 - * 2.4.20 *issue_asset*
 - * 2.4.21 *reserve_asset*

* 2.4.22 *override_transfer*

18.1 Wallet Connection Method

18.1.1 Download Method

Testnet wallet address

Official web wallet address

18.1.2 Starting Method

Using wallet:

It can be through the interactive command line, or through the websocket and http interface. Refer to [Transaction Instructions](#)

```
./yoyow_client -s ws://127.0.0.1:8090/ -r 0.0.0.0:8091 -H 127.0.0.1:8093
```

Note:

Use `-s` to specify the IP **and** port of the connected node program;
 Use the `-r` option to **open** a websocket interface;
 Use the `-H` option to **open** an HTTP-RPC service **for** easy access by our other programs. `↪`
`↪`For example, the script program **for** processing deposit/withdrawal separately.
 Yoyow_node will only listen to the RPC port after the synchronization **is** completed, `↪`
`↪`so please be patient **and** wait **for** the yoyow_node synchronization to complete.
 You can start multiple clients to connect to the same yoyow_node. But please be `↪`
`↪`careful **not** to use the same `-H`, the monitor will fail because the port **is** occupied.

The native address is used in the test cases in this document.

websocket interface address `ws://localhost:8091`

http rpc interface address `http://localhost:8093`

Connection Method

1. Use `wscat` to connect to websocket interface:

```
wscat -c ws://localhost:8091
```

2. Use `curl` to connect to websocket interface:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "get_accounts_by_uid",  
↪ [{"250926091"}]], "id": 1}' http://localhost:8091
```

3. Use `curl` to connect http interface:

```
curl --data '{"method": "call", "params": [0, "get_accounts_by_uid", [{"250926091"}]],  
↪ "id": 1}' http://localhost:8093
```

Remarks: The difference between websocket and http interface: websocket interface can also use `curl` to get data, and it will follow jsonrpc format, and the requested and returned json data must carry `{"jsonrpc": "2.0"}`. The interface of http does not need to carry the label of `{"jsonrpc": "2.0"}`.

18.2 2.1 Tools API

18.2.1 2.1.1 calculate_account_uid

Give a number, calculate the corresponding account uid

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

n number

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{{"id":1, "method":"call", "params":[0,"calculate_account_uid",[12]]}}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "calculate_account_uid",
↪ [12]], "id": 1}' http://localhost:8093
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": 3106
}
```

18.2.2 2.1.2 suggest_brain_key

Randomly generate a brain key and derive a pair of public and private keys based on the brain key

Supported format

JSON

Request Method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

null

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0,"suggest_brain_key",[]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "suggest_brain_key", []],
↵ "id": 1}' http://localhost:8091
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "brain_priv_key": "DERIC GIANTRY ALODY TELAR TIRR BOILER BULLIT REACTOR BANISH_
↪FLOCCUS SLIPPER PELANOS WEALTHY SOLE RESCRUB RELIMIT",
    "wif_priv_key": "5JXK8jhtJM8jKXcpBHeWahzkfZ9c7ske31TkMR7eMeqluWirYVD",
    "pub_key": "YYW7jcmGpu6KEUE352VtGB9PTo38Nut5qxXitfSgG6cDmAvxz2yin"
  }
}
```

18.2.3 2.1.3 get_transaction_id

Return the tx id (transaction ID, or transaction hash) of a given transaction

Supported format

JSON

Request Method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

trx Complete transaction in JSON format

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0,"get_transaction_id", [{"operations": [[0, {"fee": {
↪ "total": {"amount":100000, "asset_id":0}, "from":250926091, "to":223331844, "amount": {
↪ "amount":100000, "asset_id":0}, "extensions": {}}]]]]}]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "suggest_brain_key", [{"
↪ "operations": [[0, {"fee": {"total": {"amount":100000, "asset_id":0}, "from":250926091,
↪ "to":223331844, "amount": {"amount":100000, "asset_id":0}, "extensions": {}}]]}], "id":
↪ 1}' http://localhost:8091
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": "5ea3a6ee9f030472f83fb436836b602a3a5ed6a5"
}
```

18.3 2.2 Queries API

18.3.1 2.2.1 get_account

Get basic account information

Supported format

JSON

Request Method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

account_name_or_id uid or account nickname, for example: “250926091”

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0,"get_account",[250926091]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params":[0,"get_account",
↪[250926091]], "id": 1}' http://localhost:8091
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "id": "1.2.1378",
    "uid": 250926091,
    "name": "yoyo250926091",
    "owner": {
      "weight_threshold": 1,
      "account_uid_auths": [],
      "key_auths": [
        [
          "YYW7UoNSEQAUPvnvoBRVKyPAD9845esnpiK6MgHinsn5yqr5UgT5W",
          1
        ]
      ]
    },
    "active": {
      "weight_threshold": 1,
      "account_uid_auths": [],
      "key_auths": [
        [
          "YYW6fU7Th8uESW9FZnpkhYaTUwtSvn3f2TQxFVA3ef2SSiwdZES71",
          1
        ]
      ]
    },
    "secondary": {
      "weight_threshold": 1,
      "account_uid_auths": [],
      "key_auths": [
        [
          "YYW5eDSFYeiqyFRajfPP8tTZM7mfUeyc7H65zmnHtDW4SQJdwqTBD",
          1
        ]
      ]
    }
  },
}
```

(continues on next page)

(continued from previous page)

```
"memo_key": "YYW7SpC4QLY1LRRxFQ2hbYHdAyQo88L8qnPJcDJkiRMugcnFGUGvo",
"reg_info": {
  "registrar": 206336051,
  "referrer": 25997,
  "registrar_percent": 0,
  "referrer_percent": 0,
  "allowance_per_article": {
    "amount": 0,
    "asset_id": 0
  },
  "max_share_per_article": {
    "amount": 0,
    "asset_id": 0
  },
  "max_share_total": {
    "amount": 0,
    "asset_id": 0
  },
  "buyout_percent": 10000
},
"can_post": true,
"can_reply": false,
"can_rate": false,
"is_full_member": true,
"is_registrar": false,
"is_admin": false,
"create_time": "2018-04-03T08:21:00",
"last_update_time": "2018-04-03T08:21:00",
"active_data": "{}",
"secondary_data": "{}",
"statistics": "2.5.1378"
}
}
```

18.3.2 2.2.2 get_full_account

Get account details

Supported format

JSON

Request Method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

account_name_or_id uid or account nickname, for example: "250926091"

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0,"get_full_account",["250926091"]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "get_full_account", [{"↵"250926091"}]]}, {"id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "account": {
      "id": "1.2.1378",
      "uid": 250926091,
      "name": "yoyo250926091",
      "owner": {
        "weight_threshold": 1,
        "account_uid_auths": [],
        "key_auths": [
          [
            "YYW7UoNSEQAUPvnvoBRVKyPAD9845esnpiK6MgHinsn5yqr5UgT5W",
            1
          ]
        ]
      },
    },
    "active": {
      "weight_threshold": 1,
      "account_uid_auths": [],
      "key_auths": [
        [
          "YYW6fU7Th8uESW9FZnpkhYaTUwtSvn3f2TQxFVA3ef2SSiwdZES71",
          1
        ]
      ]
    }
  }
}
```

(continues on next page)

(continued from previous page)

```

    ]
  ],
  "secondary": {
    "weight_threshold": 1,
    "account_uid_auths": [],
    "key_auths": [
      [
        "YYW5eDSFYeiqyFRajfPP8tTZM7mfUeyc7H65zmnHtDW4SQJdwqTBD",
        1
      ]
    ]
  },
  "memo_key": "YYW7SpC4QLY1LRRxFQ2hbYHdAyQo88L8qnPJcDJkiRMugcnFGUGvo",
  "reg_info": {
    "registrar": 206336051,
    "referrer": 25997,
    "registrar_percent": 0,
    "referrer_percent": 0,
    "allowance_per_article": {
      "amount": 0,
      "asset_id": 0
    },
    "max_share_per_article": {
      "amount": 0,
      "asset_id": 0
    },
    "max_share_total": {
      "amount": 0,
      "asset_id": 0
    },
    "buyout_percent": 10000
  },
  "can_post": true,
  "can_reply": false,
  "can_rate": false,
  "is_full_member": true,
  "is_registrar": false,
  "is_admin": false,
  "create_time": "2018-04-03T08:21:00",
  "last_update_time": "2018-04-03T08:21:00",
  "active_data": "{}",
  "secondary_data": "{}",
  "statistics": "2.5.1378"
},
"statistics": {
  "id": "2.5.1378",
  "owner": 250926091,
  "total_ops": 11,
  "removed_ops": 0,
  "prepaid": 0,
  "csaf": 4200683,
  "core_balance": 1098850704,
  "core_leased_in": 0,
  "core_leased_out": 0,
  "average_coins": 1099970604,
  "average_coins_last_update": "2018-04-12T12:56:00",

```

(continues on next page)

(continued from previous page)

```
"coin_seconds_earned": "136484730731520",
"coin_seconds_earned_last_update": "2018-04-12T12:56:00",
"total_witness_pledge": 0,
"releasing_witness_pledge": 0,
"witness_pledge_release_block_number": 4294967295,
"last_witness_sequence": 0,
"uncollected_witness_pay": 0,
"witness_last_confirmed_block_num": 0,
"witness_last_aslot": 0,
"witness_total_produced": 0,
"witness_total_missed": 0,
"witness_last_reported_block_num": 0,
"witness_total_reported": 0,
"total_committee_member_pledge": 0,
"releasing_committee_member_pledge": 0,
"committee_member_pledge_release_block_number": 4294967295,
"last_committee_member_sequence": 0,
"can_vote": true,
"is_voter": false,
"last_voter_sequence": 1,
"last_platform_sequence": 1,
"total_platform_pledge": 1000000000,
"releasing_platform_pledge": 0,
"platform_pledge_release_block_number": 4294967295,
"last_post_sequence": 0
},
"csaf_leases_in": [],
"csaf_leases_out": [],
"witness_votes": [],
"committee_member_votes": []
}
}
```

18.3.3 2.2.3 get_relative_account_history

Get the account history

Supported format

JSON

Request Method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

account can be uid or account nickname

op_type the type of limited operation; see the types of operation. When the value is null, all operation types are returned; when 0, all transfer operations are available.

start query start numbersequence number

limit Return the total number of results

end When the value is 0, the most recent operation record is available.

The number of returned results will be in the end - start range; if the limit value is smaller than end - start, the latest operation record that satisfies the limit condition is returned. The returned results are sorted in the way that the latest ones are returned first.

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0,"get_relative_account_history",["250926091",
↪null,10,10,0]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params":[0,"get_relative_account_
↪history",["250926091",null,10,10,0]], "id": 1}' http://localhost:8091
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    {
      "memo": "",
      "description": "Transfer 1.20000 YOYO from 250926091 to 209414065 (Fee: 0.
↪20000 YOYO)",
      "sequence": 11,
      "op": {
        "id": "1.12.46722",
        "op": [
```

(continues on next page)

(continued from previous page)

```

    0,
    {
      "fee": {
        "total": {
          "amount": 20000,
          "asset_id": 0
        },
        "options": {
          "from_csaf": {
            "amount": 20000,
            "asset_id": 0
          }
        }
      },
      "from": 250926091,
      "to": 209414065,
      "amount": {
        "amount": 120000,
        "asset_id": 0
      }
    }
  ],
  "result": [
    0,
    {}
  ],
  "block_timestamp": "2018-04-12T12:56:21",
  "block_num": 5946192,
  "trx_in_block": 0,
  "op_in_trx": 0,
  "virtual_op": 690
}
},
{
  "memo": "",
  "description": "Transfer 10 YOYO from 250926091 to 209414065 (Fee: 0.20000_
↪YOYO)",
  "sequence": 10,
  "op": {
    "id": "1.12.46721",
    "op": [
      0,
      {
        "fee": {
          "total": {
            "amount": 20000,
            "asset_id": 0
          },
          "options": {
            "from_csaf": {
              "amount": 20000,
              "asset_id": 0
            }
          }
        }
      }
    ],
    "from": 250926091,
    "to": 209414065,

```

(continues on next page)

(continued from previous page)

```
        "amount": {
          "amount": 1000000,
          "asset_id": 0
        }
      ],
      "result": [
        0,
        {}
      ],
      "block_timestamp": "2018-04-12T12:55:57",
      "block_num": 5946184,
      "trx_in_block": 0,
      "op_in_trx": 0,
      "virtual_op": 689
    }
  ]
}
```

18.3.4 2.2.4 list_account_balances

Get the account balance

Supported format

JSON

Request Method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

account uid or account nickname, for example: "250926091"

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0,"list_account_balances",["250926091"]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "list_account_balances",
↳ ["250926091"]], "id": 1}' http://localhost:8091
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    {
      "amount": 1098850704,
      "asset_id": 0
    }
  ]
}
```

18.3.5 2.2.5 list_accounts_by_name

Find the account UID by name

Supported format

JSON

Request Method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

lowerbound Start the query with this as the starting name, set it to the empty string and start from the beginning.

limit Return quantity limit, up to 1001

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0,"list_accounts_by_name",["yoyo",10]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "list_accounts_by_name",
↪ ["yoyo", 10]], "id": 1}' http://localhost:8091
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    [
      "yoyo10007071",
      10007071
    ],
    [
      "yoyo100090928",
      100090928
    ],
    [
      "yoyo100361976",
      100361976
    ],
    [
      "yoyo100459405",
      100459405
    ],
    [
      "yoyo100501159",
      100501159
    ],
    [
      "yoyo100583445",
      100583445
    ],
  ]
}
```

(continues on next page)

(continued from previous page)

```
"yoyo100603302",
100603302
],
[
  "yoyo100735531",
  100735531
],
[
  "yoyo10124233",
  10124233
],
[
  "yoyo101530854",
  101530854
]
]
}
```

18.3.6 2.2.6 get_witness

Get witness information.

Supported format

JSON

Request Method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

owner_account The parameter can be uid or an account nickname.

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091  
{ "id":1, "method":"call", "params":[0,"get_witness",["132826789"]] }
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "get_witness", [  
  ↪ "132826789"]], "id": 1}' http://localhost:8091
```

Return results

```
{  
  "id": 1,  
  "jsonrpc": "2.0",  
  "result": {  
    "id": "1.5.31",  
    "account": 132826789,  
    "name": "yoyo132826789",  
    "sequence": 1,  
    "is_valid": true,  
    "signing_key": "YYW1111111111111111111111111111111111111111111114T1Anm",  
    "pledge": "7500000000",  
    "pledge_last_update": "2017-09-05T11:39:03",  
    "average_pledge": "7500000000",  
    "average_pledge_last_update": "2017-09-06T12:05:36",  
    "average_pledge_next_update_block": 4294967295,  
    "total_votes": 1023267564,  
    "by_pledge_position": "0",  
    "by_pledge_position_last_update": "0",  
    "by_pledge_scheduled_time": "45370982250075664161773192435",  
    "by_vote_position": "0",  
    "by_vote_position_last_update": "0",  
    "by_vote_scheduled_time": "332544857826054970738151567847",  
    "last_confirmed_block_num": 8168,  
    "last_aslot": 8599,  
    "total_produced": 25,  
    "total_missed": 0,  
    "url": ""  
  }  
}
```

18.3.7 2.2.7 list_witnesses

Query the list of the token age rental (lending) of the specified lender.

Results are sorted by borrower uid from small to large.

Supported format

JSON

Request Method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

lower_bound Start the query with this as the starting uid, set it to 0 and start from the beginning.

limit Return quantity limit, up to 101

order_by Sort type. The value range is [0, 1, 2]. 0: Sort by uid from big to small; 1: Sort by number of votes; 2: Sort by collateral amount.

Precautions

The interface uses a pagination design. To get all the witnesses, you can cycle through them until the number of witnesses returned is less than the limit.

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0,"list_witnesses",["132"]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "list_witnesses", ["132", "↵"]], "id": 1}' http://localhost:8091
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    {
      "id": "1.5.1",
      "account": 25997,
      "name": "init1",
      "sequence": 1,

```

(continues on next page)

(continued from previous page)

```

    "is_valid": true,
    "signing_key": "YYW71suPihtG7jJAGiVBCKd63ECHYebQaPa894oy3r54zk3eMlitt",
    "pledge": 1000000000,
    "pledge_last_update": "2017-09-12T21:02:45",
    "average_pledge": 1000000000,
    "average_pledge_last_update": "2017-09-13T21:20:30",
    "average_pledge_next_update_block": 4294967295,
    "total_votes": 0,
    "by_pledge_position": "0",
    "by_pledge_position_last_update": "0",
    "by_pledge_scheduled_time": "340282366580656096882718510549",
    "by_vote_position": "0",
    "by_vote_position_last_update": "0",
    "by_vote_scheduled_time": "340282366920938463463374607431768211455",
    "last_confirmed_block_num": 5994633,
    "last_aslot": 6366418,
    "total_produced": 518458,
    "total_missed": 32186,
    "url": ""
  },
  {
    "id": "1.5.2",
    "account": 26264,
    "name": "init2",
    "sequence": 1,
    "is_valid": true,
    "signing_key": "YYW71suPihtG7jJAGiVBCKd63ECHYebQaPa894oy3r54zk3eMlitt",
    "pledge": 1000000000,
    "pledge_last_update": "2017-09-12T21:02:51",
    "average_pledge": 1000000000,
    "average_pledge_last_update": "2017-09-13T21:20:36",
    "average_pledge_next_update_block": 4294967295,
    "total_votes": 0,
    "by_pledge_position": "0",
    "by_pledge_position_last_update": "0",
    "by_pledge_scheduled_time": "340282366580656096882718510549",
    "by_vote_position": "0",
    "by_vote_position_last_update": "0",
    "by_vote_scheduled_time": "340282366920938463463374607431768211455",
    "last_confirmed_block_num": 5994632,
    "last_aslot": 6366417,
    "total_produced": 518439,
    "total_missed": 32198,
    "url": ""
  },
  {
    "id": "1.5.3",
    "account": 26460,
    "name": "init3",
    "sequence": 1,
    "is_valid": true,
    "signing_key": "YYW71suPihtG7jJAGiVBCKd63ECHYebQaPa894oy3r54zk3eMlitt",
    "pledge": 1000000000,
    "pledge_last_update": "2017-09-12T21:02:54",
    "average_pledge": 1000000000,
    "average_pledge_last_update": "2017-09-13T21:20:39",
    "average_pledge_next_update_block": 4294967295,

```

(continues on next page)

(continued from previous page)

```
"total_votes": 0,
"by_pledge_position": "0",
"by_pledge_position_last_update": "0",
"by_pledge_scheduled_time": "340282366580656096882718510549",
"by_vote_position": "0",
"by_vote_position_last_update": "0",
"by_vote_scheduled_time": "340282366920938463463374607431768211455",
"last_confirmed_block_num": 5994636,
"last_aslot": 6366421,
"total_produced": 518427,
"total_missed": 32161,
"url": ""
},
{
  "id": "1.5.4",
  "account": 26861,
  "name": "init4",
  "sequence": 1,
  "is_valid": true,
  "signing_key": "YYW71suPihtG7jJAGiVBCKd63ECHYebQaPa894oy3r54zk3eMlitt",
  "pledge": 1000000000,
  "pledge_last_update": "2017-09-12T21:03:00",
  "average_pledge": 1000000000,
  "average_pledge_last_update": "2017-09-13T21:20:45",
  "average_pledge_next_update_block": 4294967295,
  "total_votes": 0,
  "by_pledge_position": "0",
  "by_pledge_position_last_update": "0",
  "by_pledge_scheduled_time": "340282366580656096882718510549",
  "by_vote_position": "0",
  "by_vote_position_last_update": "0",
  "by_vote_scheduled_time": "340282366920938463463374607431768211455",
  "last_confirmed_block_num": 5994640,
  "last_aslot": 6366425,
  "total_produced": 518441,
  "total_missed": 32137,
  "url": ""
},
{
  "id": "1.5.5",
  "account": 27027,
  "name": "init5",
  "sequence": 1,
  "is_valid": true,
  "signing_key": "YYW71suPihtG7jJAGiVBCKd63ECHYebQaPa894oy3r54zk3eMlitt",
  "pledge": 1000000000,
  "pledge_last_update": "2017-09-12T21:05:15",
  "average_pledge": 1000000000,
  "average_pledge_last_update": "2017-09-13T21:23:00",
  "average_pledge_next_update_block": 4294967295,
  "total_votes": 0,
  "by_pledge_position": "0",
  "by_pledge_position_last_update": "0",
  "by_pledge_scheduled_time": "340282366580656096882718510549",
  "by_vote_position": "0",
  "by_vote_position_last_update": "0",
  "by_vote_scheduled_time": "340282366920938463463374607431768211455",
```

(continues on next page)

(continued from previous page)

```
"last_confirmed_block_num": 5994639,  
"last_aslot": 6366424,  
"total_produced": 518387,  
"total_missed": 32190,  
"url": ""  
}  
]  
}
```

18.3.8 2.2.8 get_committee_member

Get the committee member information

Supported format

JSON

Request Method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

owner_account Uid or account nickname. For example: "25997"

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091  
{ "id":1, "method":"call", "params":[0,"get_committee_member",["25997"]]
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "get_committee_member", [↵
↵ "25997"]], "id": 1}' http://localhost:8091
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "id": "1.4.0",
    "account": 25997,
    "name": "init1",
    "sequence": 1,
    "is_valid": true,
    "pledge": 0,
    "total_votes": 567814657,
    "url": ""
  }
}
```

18.3.9 2.2.9 list_committee_members

List the current valid committee candidates

Supported format

JSON

Request Method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

lower_bound Start the query with this as the starting uid, set it to 0 and start from the beginning.

limit Return quantity limit, up to 101

order_by Sort type. Value range is [0, 1, 2]. 0: Sort by uid from big to small; 1: Sort by number of votes; 2: Sort by collateral amount.

Precautions

The interface adopts a pagination design. To obtain all the committees, you can call them cyclically until the number of committee members returned is less than the limit.

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0,"list_committee_members",[0,5,1]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "list_committee_members", [0, 5, 1]], "id": 1}' http://localhost:8091
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    {
      "id": "1.4.0",
      "account": 25997,
      "name": "init1",
      "sequence": 1,
      "is_valid": true,
      "pledge": 0,
      "total_votes": 567814657,
      "url": ""
    },
    {
      "id": "1.4.1",
      "account": 26264,
      "name": "init2",
      "sequence": 1,
      "is_valid": true,
      "pledge": 0,
      "total_votes": 0,
      "url": ""
    },
    {
      "id": "1.4.2",
      "account": 26460,
      "name": "init3",
      "sequence": 1,
      "is_valid": true,
      "pledge": 0,
      "total_votes": 0,
      "url": ""
    },
    {
      "id": "1.4.3",
```

(continues on next page)

(continued from previous page)

```
"account": 26861,
"name": "init4",
"sequence": 1,
"is_valid": true,
"pledge": 0,
"total_votes": 0,
"url": ""
},
{
  "id": "1.4.4",
  "account": 27027,
  "name": "init5",
  "sequence": 1,
  "is_valid": true,
  "pledge": 0,
  "total_votes": 0,
  "url": ""
}
]
}
```

18.3.10 2.2.10 list_committee_proposals

List all the committee proposals that have not been successfully implemented, including those that are being voted on, have been voted through but have not yet reached the execution time.

Supported format

JSON

Request Method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

null

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "list_committee_proposals", []]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params":[0, "list_committee_
↵proposals", []], "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": []
}
```

18.3.11 2.2.11 get_platform_count

Get the total number of platforms on the network

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

null

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "get_platform_count", []]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params":[0, "get_platform_count", ↵
↵[]], "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": 6
}
```

18.3.12 2.2.12 get_platform

Get platform object information according to the platform owner account

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

owner_account Platform owner account

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "get_platform", ["250926091"]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "update_committee_
↪account", [0, "get_platform", ["250926091"]], "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "id": "1.6.3",
    "owner": 250926091,
    "name": "NoPlatform",
    "sequence": 1,
    "is_valid": true,
    "total_votes": 0,
    "url": "www.example2.com",
    "pledge": 1000000000,
    "pledge_last_update": "2018-04-03T09:30:48",
    "average_pledge": 396825,
    "average_pledge_last_update": "2018-04-03T09:34:48",
    "average_pledge_next_update_block": 5684416,
    "extra_data": "{}",
    "create_time": "2018-04-03T09:30:48",
    "last_update_time": "2018-04-03T09:34:48"
  }
}
```

18.3.13 2.2.13 list_platforms

Query by platform owner to list the current valid platforms

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

lower_bound Start the query with this as the starting uid, set it to 0 and start from the beginning.

limit Return quantity limit, up to 100

order_by Sort type. The value range is [0, 1, 2]. 0:Sort by uid from big to small; 1: Sort by number of votes; 2: Sort by collateral amount.

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "list_platforms", [0,5,1]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "list_platforms", [0,
↵ "list_platforms", [0,5,1]], "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    {
      "id": "1.6.0",
      "owner": 217895094,
      "name": "test-yoyow",
      "sequence": 1,
      "is_valid": true,
      "total_votes": 0,
      "url": "http://demo.yoyow.org/",
      "pledge": 1000000000,
      "pledge_last_update": "2018-02-10T01:03:57",
      "average_pledge": 176601774,
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```

"average_pledge_last_update": "2018-02-11T06:49:12",
"average_pledge_next_update_block": 4562164,
"extra_data": "{\"login\":\"http://demo.yoyow.org:3000/authLogin\"}",
"create_time": "2018-02-10T01:03:57",
"last_update_time": "2018-02-11T06:49:12"
},
{
  "id": "1.6.5",
  "owner": 223331844,
  "name": "baidu",
  "sequence": 1,
  "is_valid": true,
  "total_votes": 0,
  "url": "",
  "pledge": 1000000000,
  "pledge_last_update": "2018-04-16T02:52:36",
  "average_pledge": 0,
  "average_pledge_last_update": "2018-04-16T02:52:36",
  "average_pledge_next_update_block": 6050467,
  "extra_data": "",
  "create_time": "2018-04-16T02:52:36",
  "last_update_time": "1970-01-01T00:00:00"
},
{
  "id": "1.6.4",
  "owner": 224006453,
  "name": "dwgMarket",
  "sequence": 1,
  "is_valid": true,
  "total_votes": 0,
  "url": "www.cad1688.com",
  "pledge": 1000000000,
  "pledge_last_update": "2018-04-04T08:38:24",
  "average_pledge": 0,
  "average_pledge_last_update": "2018-04-04T08:38:24",
  "average_pledge_next_update_block": 5712088,
  "extra_data": "{}",
  "create_time": "2018-04-04T08:38:24",
  "last_update_time": "1970-01-01T00:00:00"
}
]
}

```

18.3.14 2.2.14 get_asset

Return the asset details based on the given asset code or id

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

asset_name_or_id Asset symbol or asset id

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "get_asset", [ 3]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "get_asset", [ 3]], "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "id": "1.3.3",
  "asset_id": 3,
  "symbol": "WOWO",
  "precision": 4,
  "issuer": 223331844,
  "options": {
    "max_supply": "1000000000000000",
    "market_fee_percent": 0,
    "max_market_fee": "1000000000000000",
    "issuer_permissions": 79,
    "flags": 0,
    "whitelist_authorities": [],
    "blacklist_authorities": [],
    "whitelist_markets": [],
    "blacklist_markets": [],
    "description": ""
  },
  "dynamic_asset_data_id": "2.3.3"
}
```

18.3.15 2.2.15 list_assets

Query asset details by page

The returned results are sorted by the ASCII code order of the asset code.

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

lower_bound_symbol Start the query with this as the starting code, sorted by the ASCII code of the asset code.

limit Return quantity limit, up to 101

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "list_assets", ["YOY", 4]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "list_assets", ["YOY", ↵
↵4]], "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    {
      "id": "1.3.91",
      "asset_id": 91,
      "symbol": "YOYES",
      "precision": 2,
      "issuer": 215074501,
      "options": {
        "max_supply": 1200,
        "market_fee_percent": 0,
        "max_market_fee": 1200,
        "issuer_permissions": 79,
        "flags": 0,
        "whitelist_authorities": [],
        "blacklist_authorities": [],
        "whitelist_markets": [],
        "blacklist_markets": [],
        "description": ""
      },
      "dynamic_asset_data_id": "2.2.91",
      "dynamic_asset_data": {
        "id": "2.2.91",
        "asset_id": 91,
        "current_supply": 0,
        "accumulated_fees": 0
      }
    },
    {
      "id": "1.3.130",
      "asset_id": 130,
      "symbol": "YOYIO",
      "precision": 2,
      "issuer": 254208024,
      "options": {
        "max_supply": 1258000000,
        "market_fee_percent": 0,
        "max_market_fee": 1258000000,
        "issuer_permissions": 79,
        "flags": 0,
        "whitelist_authorities": [],
        "blacklist_authorities": [],
        "whitelist_markets": [],
        "blacklist_markets": [],
        "description": ""
      },
      "dynamic_asset_data_id": "2.2.130",
      "dynamic_asset_data": {
        "id": "2.2.130",
        "asset_id": 130,
        "current_supply": 1258000000,
        "accumulated_fees": 0
      }
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```

    },
    {
      "id": "1.3.0",
      "asset_id": 0,
      "symbol": "YOYO",
      "precision": 5,
      "issuer": 1264,
      "options": {
        "max_supply": "2000000000000000",
        "market_fee_percent": 0,
        "max_market_fee": "1000000000000000",
        "issuer_permissions": 0,
        "flags": 0,
        "whitelist_authorities": [],
        "blacklist_authorities": [],
        "whitelist_markets": [],
        "blacklist_markets": [],
        "description": ""
      },
      "dynamic_asset_data_id": "2.2.0",
      "dynamic_asset_data": {
        "id": "2.2.0",
        "asset_id": 0,
        "current_supply": "106901076031525",
        "accumulated_fees": 0
      }
    },
    {
      "id": "1.3.2",
      "asset_id": 2,
      "symbol": "YOYOW",
      "precision": 5,
      "issuer": 25638,
      "options": {
        "max_supply": "10000000000000",
        "market_fee_percent": 0,
        "max_market_fee": "10000000000000",
        "issuer_permissions": 79,
        "flags": 0,
        "whitelist_authorities": [],
        "blacklist_authorities": [],
        "whitelist_markets": [],
        "blacklist_markets": [],
        "description": ""
      },
      "dynamic_asset_data_id": "2.2.2",
      "dynamic_asset_data": {
        "id": "2.2.2",
        "asset_id": 2,
        "current_supply": 0,
        "accumulated_fees": 0
      }
    }
  ]
}

```

18.4 2.3 Wallet/Private Key Management API

18.4.1 2.3.1 save_wallet_file

Save the wallet file and it will be saved to the yoyo_client executable folder

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

Wallet needs to be in unlock state.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

wallet_filename String, the name of the backup file

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "save_wallet_file", ["t3.json"]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "save_wallet_file", [
↪ "t3.json"]], "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": null
}
```

18.4.2 2.3.2 set_password

Set the wallet password

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

Wallet needs to be in new or unlocked state.

The new state exists when it is the first time that the wallet has been run, and the password state has not been set.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

password Password string, for example: "1234"

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "set_password", ["1234"]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "set_password", ["1234
↵"]], "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": null
}
```

18.4.3 2.3.3 unlock

Unlock the wallet

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

The wallet is in locked state.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

password Password string, for example: “1234”

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "unlock", ["1234"]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "unlock", ["1234"]],
  ↪ "id": 1}'
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": null
}
```

18.4.4 2.3.4 lock

Lock the wallet

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

null

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "lock", []]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "lock", []], "id": 1}'
↪ http://localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": null
}
```

18.4.5 2.3.5 import_key

Import a private key into your wallet and assign a related account. The private key and account number do not have to be associated.

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

Wallet needs to be in unlocked state.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

account_name_or_id Account uid or nickname

wif_key Private key string

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "lock", []]}{"id":1, "method":"call", "params
↪": [0, "import_key", ["250926091",
↪ "5JLaW7u3EC4vVLbTmLolXeSBGiTeRtqER1UsoLtYbFNnBafgPKG"]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "lock", []]}{"id":1,
↪ "method": "call", "params": [0, "import_key", ["250926091",
↪ "5JLaW7u3EC4vVLbTmLolXeSBGiTeRtqER1UsoLtYbFnnBafgPKG"]], "id": 1}' http://
↪ localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": true
}
```

18.4.6 2.3.6 dump_private_keys

List all private keys and corresponding public keys in the wallet

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

Wallet needs to be in unlocked state

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

null

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "dump_private_keys",[]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "dump_private_keys",
→[], "id": 1]}' http://localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    [
      "YYW5eDSFYeiqyFRajfPP8tTZM7mfUeyc7H65zmnHtDW4SQJdwqTBD",
      "5HvABsnYU1U7misWHq9mc6mE8QovBiy8H5rVZc3zKztgZsPffMB"
    ],
    [
      "YYW6fU7Th8uESW9FZnpkhYaTUwtSvn3f2TQxFVA3ef2SSiwdZES71",
      "5JLaW7u3EC4vVLbTmLolXeSBGiTeRtqER1UsoLtYbFnnBafgPKG"
    ]
  ]
}
```

18.4.7 2.3.7 list_my_accounts_cached

List the information about all cached accounts in the wallet file (the account specified when the private key was imported)

Note: This cached information is not necessarily synchronized with the data on the chain. To sync, reopen the wallet file.

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

Wallet needs to be in unlocked state.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

null

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "list_my_accounts_cached", []]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "list_my_accounts_
↪cached", []], "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": [
    {
      "id": "1.2.1378",
      "uid": 250926091,
      "name": "yoyo250926091",
      "owner": {
        "weight_threshold": 1,
        "account_uid_auths": [],
        "key_auths": [
          [
            "YYW7UoNSEQAUPvnvoBRVKyPAD9845esnpiK6MgHinsn5yqr5UgT5W",
            1
          ]
        ]
      },
      "active": {
        "weight_threshold": 1,
        "account_uid_auths": [],
        "key_auths": [
          [
            "YYW6fU7Th8uESW9FZnpkhYaTUwtSvn3f2TQxFVA3ef2SSiwdZES71",
            1
          ]
        ]
      }
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```

    ]
  ],
  "secondary": {
    "weight_threshold": 1,
    "account_uid_auths": [],
    "key_auths": [
      [
        "YYW5eDSFYeiqyFRajfPP8tTZM7mfUeyc7H65zmnHtDW4SQJdwqTBD",
        1
      ]
    ]
  },
  "memo_key": "YYW7SpC4QLY1LRRxFQ2hbYHdAyQo88L8qnPJcDJkiRMugcnFGUGvo",
  "reg_info": {
    "registrar": 206336051,
    "referrer": 25997,
    "registrar_percent": 0,
    "referrer_percent": 0,
    "allowance_per_article": {
      "amount": 0,
      "asset_id": 0
    },
    "max_share_per_article": {
      "amount": 0,
      "asset_id": 0
    },
    "max_share_total": {
      "amount": 0,
      "asset_id": 0
    },
    "buyout_percent": 10000
  },
  "can_post": true,
  "can_reply": false,
  "can_rate": false,
  "is_full_member": true,
  "is_registrar": false,
  "is_admin": false,
  "create_time": "2018-04-03T08:21:00",
  "last_update_time": "2018-04-03T08:21:00",
  "active_data": "{}",
  "secondary_data": "{}",
  "statistics": "2.5.1378"
}
]
}

```

18.5 2.4 Operations/Transactions API

The following operations involve key authority. You need to import the relevant private keys. At the same time, ensure that the wallet is in unlocked state.

18.5.1 2.4.1 transfer

Query the platforms according to the uid list

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

It needs the active key of the transferrer.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

from Transferrer (UID or nickname)

to Transferee (UID or nickname)

amount Amount, if the amount is a decimal, it is recommended to use the string to pass the parameter

asset_symbol token type; the asset type is currently only “YOYO”.

memo Memo (use an empty string “” if without a memo)

broadcast Whether to broadcasttrue or false

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "transfer",[250926091, 209414065, "10", "YOYO",
↪ "feho", true]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params":[0, "transfer",[250926091, ↪
↪209414065, "10", "YOYO", "feho", true]], "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "ref_block_num": 57170,
    "ref_block_prefix": 852086171,
    "expiration": "2018-04-15T03:18:33",
    "operations": [
      [
        0,
        {
          "fee": {
            "total": {
              "amount": 20898,
              "asset_id": 0
            },
            "options": {
              "from_csaf": {
                "amount": 20898,
                "asset_id": 0
              }
            }
          },
          "from": 250926091,
          "to": 209414065,
          "amount": {
            "amount": 1000000,
            "asset_id": 0
          },
          "memo": {
            "from": "YYW7SpC4QLY1LRRxFQ2hbYHdAyQo88L8qnPJcDJkiRMugcnFGUGvo",
            "to": "YYW8PK8NtXW6JjtxqPV8QTgw4ejPEg4FgVeNVlmaZDGzzNoEbgmr2",
            "nonce": "7783743918290282490",
            "message": "4468a7f3a5ac7fbf8125856381673030"
          }
        }
      ]
    ],
    "signatures": [
      ↪ "1f0a075215760089cf879b67ee6ba0aaaffa9408cd48c9040eee562909a8d67f5f7bbbb6401aabc69c00cd5d212f65b412",
      ↪ ""
    ]
  }
}
```

18.5.2 2.4.2 create_witness

Create a witness

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

It needs the active key of the witness owner.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

- owner_account** AccountUID or nickname
- block_signing_key** Block production signature public keys
- pledge_amount** Collateral amount
- pledge_asset_symbol** Collateral token typeYOYO
- url** Introduction link
- broadcast** Whether to broadcast

Wherein: if the signature public key is YYW1111111111111111111111111111111114T1Anm, it means temporary offline.

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "create_witness", ["223331844",
↪ "YYW1111111111111111111111111111111114T1Anm", "1000000", "YOYO", "http://www.yoyow.
↪ org", true]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params":[0, "create_witness", [
↪ "223331844", "YYW1111111111111111111111111111111114T1Anm", "100", "YOYO", "http://
↪ www.yoyow.org", true]], "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "ref_block_num": 58457,
  "ref_block_prefix": 2372452101,
  "expiration": "2018-05-03T11:28:36",
  "operations": [[
    13,{
      "fee": {
        "total": {
          "amount": 100000000,
          "asset_id": 0
        }
      },
      "account": 223331844,
      "block_signing_key": "YYW11111111111111111111111111111114T1Anm",
      "pledge": {
        "amount": "100000000000",
        "asset_id": 0
      },
      "url": "http://www.yoyow.org"
    }
  ]
},
"signatures": [
↵"202857a37e91889a1c6124a2e3405eff00647b315aa55db7989334e187a5a92c1f0cb4bb00531fa525e53f26403e8bd32
↵"
]
}
```

18.5.3 2.4.3 update_witness

Modify witness information

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

It needs the active key of the witness owner.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

witness_account AccountUID or nickname

block_signing_key New block production signature public key; enter null if without modification

pledge_amount New collateral amount; enter null if without modification

pledge_asset_symbol New collateral token type (YOYO); enter null if without modification

url New intro link; enter null if without modification

broadcast Whether to broadcast

Among them, the amount of the collateral and the token type must appear at the same time or not at the same time, the current token type can only be YOYO.

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "update_witness", ["223331844", null,"100345",
↪"YOYO", null, true]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "update_witness", [
↪"223331844", null,"100345", "YOYO", null, true]], "id": 1}' http://localhost:8091/
↪rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "ref_block_num": 13103,
    "ref_block_prefix": 3050749194,
    "expiration": "2018-05-04T04:17:42",
    "operations": [
      [
        14,
        {
          "fee": {
            "total": {
              "amount": 1000000,
              "asset_id": 0
            },
          },
          "options": {
            "from_csaf": {
              "amount": 1000000,
```

(continues on next page)

(continued from previous page)

```
        "asset_id": 0
      }
    },
    "account": 223331844,
    "new_pledge": {
      "amount": "10034500000",
      "asset_id": 0
    }
  ]
},
"signatures": [
  ↪ "1f6503a1e7dd15d1d9d5fe9cdadddeea39acf40071bd5621458b9abf3e0c8709f63fedfac89adc571fcc8af20fe6beb9f"
  ↪ "
  ]
}
}
```

18.5.4 2.4.4 create_committee_member

Create a committee candidate identity

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

owner_account AccountUID or nickname

pledge_amount Collateral amount

pledge_asset_symbol Collateral token typeYOYO

url into link

broadcast Whether to broadcast

Precautions

Among the assets found, only YOYO can actually be used.

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "create_committee_member", ["223331844","1000",
↪ "YOYO", "http://www.yoyow.org", true]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params":[0, "create_committee_
↪member", ["223331844","1000", "YOYO", "http://www.yoyow.org", true]], "id": 1}'_
↪http://localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "ref_block_num": 13153,
    "ref_block_prefix": 1417310192,
    "expiration": "2018-05-04T04:20:12",
    "operations": [
      [
        8,
        {
          "fee": {
            "total": {
              "amount": 10000000,
              "asset_id": 0
            }
          },
          "account": 223331844,
          "pledge": {
            "amount": 100000000,
            "asset_id": 0
          },
          "url": "http://www.yoyow.org"
        }
      ]
    ],
    "signatures": [
↪ "1f2b34fe5e2437be46d83ec2f0f4482e1b5df509131131c41eeb16e484df5e4ea96df19f82be294433bc751e84d6dcc28
↪ "
  ]
}
```

(continues on next page)

(continued from previous page)

```
}  
}
```

18.5.5 2.4.5 update_committee_member

Modify committee candidate information

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

committee_member_account AccountUID or nickname

pledge_amount New collateral amount; enter null if without modification

pledge_asset_symbol New collateral token type (YOYO); enter null if without modification

url New intro link; enter null if without modification

broadcast Whether to broadcast

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091  
{ "id":1, "method":"call", "params":[0, "update_committee_member", ["223331844", "10234  
↵", "YOYO", null, true]] }
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "update_committee_
↪account", ["250926091", "10000", "YOYO", null, true]], "id": 1}' http://
↪localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "ref_block_num": 13189,
    "ref_block_prefix": 2763581564,
    "expiration": "2018-05-04T04:22:00",
    "operations": [
      [
        9,
        {
          "fee": {
            "total": {
              "amount": 1000000,
              "asset_id": 0
            },
            "options": {
              "from_csaf": {
                "amount": 1000000,
                "asset_id": 0
              }
            }
          },
          "account": 223331844,
          "new_pledge": {
            "amount": 1023400000,
            "asset_id": 0
          }
        }
      ]
    ],
    "signatures": [
      ↪"20506ea2aadb44a57ae4bb60c71b0c2002f89410d4941ed83d3323c4bed2f883ee4d045c9a326e331b49770db32799c631
      ↪"
    ]
  }
}
```

18.5.6 2.4.6 set_voting_proxy

Set up a voting proxy

Account A sets account B as a voting proxy, and the number of votes obtained by the voting object of B is the number of valid votes of A + the number of valid votes of B. A is called the client, and B is called the proxy.

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

It needs the active key of the client.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

account_to_modify Client accountUID or nickname

voting_account Proxy accountset the proxy with a UID or nickname; null is for canceling the proxy

broadcast Whether to broadcast

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "set_voting_proxy", ["250926091", "abit", true]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params":[0, "set_voting_proxy", ["250926091", "abit", true]], "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "ref_block_num": 18995,
```

(continues on next page)

(continued from previous page)

```

"ref_block_prefix": 2835940919,
"expiration": "2018-04-16T02:06:36",
"operations": [
  [
    5,
    {
      "fee": {
        "total": {
          "amount": 100000,
          "asset_id": 0
        },
        "options": {
          "from_csaf": {
            "amount": 100000,
            "asset_id": 0
          }
        }
      },
      "voter": 223331844,
      "proxy": 250926091
    }
  ]
],
"signatures": [
  ↪ "1f793459c8c7e06e80b2b34d2d13a0fb46e5d4f839953f6fae96af16acf389b51c534c35d2f85fe5d9f8e7316b1bb6694"
  ↪ "
]
}
}

```

18.5.7 2.4.7 update_witness_votes

Witness voting

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

It needs the active key of the voter.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

voting_account AccountUID of nickname

witnesses_to_add Add a list of supported witnesses (UID or nickname)

witnesses_to_remove Remove the list of supported witnesses (UID or nickname)

broadcast Whether to broadcast

Both the `witnesses_to_add` and the `witnesses_to_remove` lists can be empty `[]`, indicating that the voting intention is refreshed.

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "update_witness_votes", ["250926091", ["abit"],
↪ [], true]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "update_witness_votes",
↪ ["250926091", ["abit"], [], true]], "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "ref_block_num": 18961,
    "ref_block_prefix": 1229162670,
    "expiration": "2018-04-16T02:04:54",
    "operations": [
      [
        15,
        {
          "fee": {
            "total": {
              "amount": 200000,
              "asset_id": 0
            },
          },
          "options": {
            "from_csaf": {
              "amount": 200000,
              "asset_id": 0
            }
          }
        }
      ]
    ],
  },
}
```

(continues on next page)

(continued from previous page)

```

        "voter": 250926091,
        "witnesses_to_add": [
            209414065
        ],
        "witnesses_to_remove": []
    }
]
],
"signatures": [
↪ "206badbed989fcf01c93a2eda807976bae29f2e95ca2dcaa83f645be6c3bffcabc178199f4e4816801643cc9ee158fc4e8"
↪ "
    ]
}
}

```

18.5.8 2.4.8 update_committee_member_votes

The committee election voting

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

It needs the active key of the voter.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

voting_account Voter account (UID or nickname)

committee_members_to_add Array; add a list of supported committee candidates (UID or nickname)

committee_members_to_remove Array; remove a list of supported committee candidates (UID or nickname)

broadcast Whether to broadcast

Both `committee_members_to_add` and `committee_members_to_remove` lists can be empty `[]`, indicating the voting intention is refreshed.

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "update_committee_member_votes", ["250926091",
↪["init1"], [], true]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params":[0, "update_committee_
↪member_votes", ["250926091", ["init1"], [], true]], "id": 1}' http://
↪localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "ref_block_num": 19152,
    "ref_block_prefix": 1139468448,
    "expiration": "2018-04-16T02:14:27",
    "operations": [
      [
        10,
        {
          "fee": {
            "total": {
              "amount": 100000,
              "asset_id": 0
            },
            "options": {
              "from_csaf": {
                "amount": 100000,
                "asset_id": 0
              }
            }
          },
          "voter": 250926091,
          "committee_members_to_add": [
            25997
          ],
          "committee_members_to_remove": []
        }
      ]
    ],
    "signatures": [
      ↪"1f35562e4301c20f293977ffe27399ccf961fc3d5c0c9d928730ed5af03af24637599e30d070032bae887d9db3201c8911
      ↪"
```

(continues on next page)

(continued from previous page)

```
    ]  
  }  
}
```

18.5.9 2.4.9 collect_csaf_with_time

To collect points, you need to specify the time parameters to collect the points accumulated to the specified time.

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

If the Collecting account is the same as Receiving account, It only needs the Secondary Key of the collector.

If the Receiving account is another account, It needs the Active Key of the collector.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

from Collecting accountUID or nickname

to Receiving accountUID or nickname

amount Collect the amount

asset_symbol Collect the token type (the token type can only be YOYO)

time Specified timefor example: "2018-04-16T02:44:00". This time is UTC time and must not be 5 minutes before the time of newly produced block on the current chain.

broadcast Whether to broadcast

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "collect_csaf_with_time", ["223331844",
↪ "223331844", "0.5", "YOYO", "2018-04-16T02:44:00" true]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params":[0, "collect_csaf_with_time
↪ ", ["223331844", "223331844", "0.5", "YOYO", "2018-04-16T02:44:00" true]], "id": 1}
↪ ' http://localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "ref_block_num": 19813,
    "ref_block_prefix": 1809327617,
    "expiration": "2018-04-16T02:47:30",
    "operations": [
      [
        6,
        {
          "fee": {
            "total": {
              "amount": 100000,
              "asset_id": 0
            },
            "options": {
              "from_csaf": {
                "amount": 100000,
                "asset_id": 0
              }
            }
          },
          "from": 223331844,
          "to": 223331844,
          "amount": {
            "amount": 50000,
            "asset_id": 0
          },
          "time": "2018-04-16T02:44:00"
        }
      ]
    ],
    "signatures": [
      ↪ "1f250855fcc4e4ef093c14990411b1cfd41f97de43447e1b6a21cbe26eb95f6c9671b7c0d5ba4365d76018d277086c34c
      ↪ ",
      ↪ "2061c58d04a7ad9f60af1f145c837f57475d3d1785754527753b1144c1bef445240faa079b5927956be10693711b392b7
      ↪ "
```

(continues on next page)

(continued from previous page)

```
}  
  }  
}
```

18.5.10 2.4.10 collect_csaf

Collect points and collect points accumulated to the current time (minutes).

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

If the Collecting account is the same as Receiving account, It only needs the Secondary Key of the collector.

If the Receiving account is another account, It needs the Active Key of the collector.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

from Collecting accountUID or nickname

to Receiving accountUID or nickname

amount Collecting the amount

asset_symbol Collecting the token type (the token type can only be YOYO)

broadcast Whether to broadcast

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "collect_csaf", ["250926091", "250926091", 1,
↪"YOYO", true]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "collect_csaf", [
↪"250926091", "250926091", 1, "YOYO", true]], "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "ref_block_num": 19646,
    "ref_block_prefix": 555752677,
    "expiration": "2018-04-16T02:39:09",
    "operations": [
      [
        6,
        {
          "fee": {
            "total": {
              "amount": 100000,
              "asset_id": 0
            },
            "options": {
              "from_csaf": {
                "amount": 100000,
                "asset_id": 0
              }
            }
          },
          "from": 250926091,
          "to": 250926091,
          "amount": {
            "amount": 100000,
            "asset_id": 0
          },
          "time": "2018-04-16T02:37:00"
        }
      ]
    ],
    "signatures": [
      ↪"203a417b25f10110d8143d7476976abbcbb3490f13432630366e5b0d1d8d7580573c8595e93109af4a55282756b8b4916a
      ↪",
      ↪"2054d3b25618ddaeeae499297a483d5490bac77f35bac7dd850645400d7f8001a2265cd997ff62db54740e9fcda52b0bbb
      ↪"
    ]
  }
}
```

18.5.11 2.4.11 create_platform

Creating platforms

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

It needs the Active key of the applier.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

owner_account Creator account

name Platform name

pledge_amount The collateral amount, currently not less than 10,000 YOYO

pledge_asset_symbol collateral token typeYOYO

url platform link

extra_data Platform additional data

broadcast Whether to broadcast

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "create_platform", ["223331844", "yoyo.club",
↪ "10000", "YOYO", "", "", true]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "create_platform", [
↪ "223331844", "yoyow.club", "10000", "YOYO", "", "", true]], "id": 1}' http://
↪ localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "ref_block_num": 19954,
    "ref_block_prefix": 1357577324,
    "expiration": "2018-04-16T02:54:33",
    "operations": [
      [
        20,
        {
          "fee": {
            "total": {
              "amount": 100007811,
              "asset_id": 0
            }
          },
          "account": 223331844,
          "pledge": {
            "amount": 1000000000,
            "asset_id": 0
          },
          "name": "baidu",
          "url": "",
          "extra_data": ""
        }
      ]
    ],
    "signatures": [
      ↪ "20534af4af03c6d4001c797dde6ac438a6b3d31c77b94cb8e4b6519e681a289c69370057de58412bb5e3ba8320ab975d3
      ↪ "
    ]
  }
}
```

18.5.12 2.4.12 update_platform

Modify platform information

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

null

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

platform_account AccountUID or nickname

name new platform name; enter null if without modification

pledge_amount New collateral amount; enter null if without modification

pledge_asset_symbol New collateral token type (YOYO); enter null if without modification

url New intro link; enter null if without modification

extra_data New platform additional data

broadcast Whether to broadcast

Note: The amount of the collateral and the token type must appear at the same time or not at the same time. The current token type can only be YOYO. If the collateral amount is 0, it means closing the platform.

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "update_platform", ["223331844", "NUUUU", null,
↵ null, "http://www.example.com", "http://www.example.com", true]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params":[0, "update_platform", [
↵ "223331844", "NUUUU", null, null, "http://www.example.com", "http://www.example.com
↵", true]], "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "ref_block_num": 20262,
```

(continues on next page)

(continued from previous page)

```
"ref_block_prefix": 1534083365,
"expiration": "2018-04-16T03:09:57",
"operations": [
  [
    21,
    {
      "fee": {
        "total": {
          "amount": 1053709,
          "asset_id": 0
        },
        "options": {
          "from_csaf": {
            "amount": 1053709,
            "asset_id": 0
          }
        }
      },
      "account": 223331844,
      "new_name": "NUUUU",
      "new_url": "http://www.example.com",
      "new_extra_data": "http://www.example.com"
    }
  ]
],
"signatures": [
  ↪ "202e8e53a7e58d4b60c7bf7b0d3f8076a6c9b8f7c472c48e61463cff68228e2cf643404bf954f1c7596deb05630942c95"
  ↪ "
  ]
}
}
```

18.5.13 2.4.13 update_platform_votes

Vote for the platforms

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

It needs the Active key of the voter.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

voting_account Voter accountUID or nickname

platforms_to_add Add a list of supported platforms (UID or nickname)

platforms_to_remove Remove a list of supported platforms (UID or nickname)

broadcast Whether to broadcast

Both platforms_to_add and platforms_to_remove lists can be empty, indicating the voting intention is refreshed.

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "update_platform_votes", ["250926091", [
↪"223331844"], [], true]}}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "update_platform_votes
↪", ["250926091", ["223331844"], [], true]], "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "ref_block_num": 20494,
    "ref_block_prefix": 3288350547,
    "expiration": "2018-04-16T03:21:33",
    "operations": [
      [
        22,
        {
          "fee": {
            "total": {
              "amount": 200000,
              "asset_id": 0
            },
            "options": {
```

(continues on next page)

(continued from previous page)

```
        "from_csaf": {
            "amount": 200000,
            "asset_id": 0
        }
    },
    "voter": 250926091,
    "platform_to_add": [
        223331844
    ],
    "platform_to_remove": []
}
],
"signatures": [
    ↪ "20274d50cf4905fe072e3257632335546c386721f2d608cf3939316f7167ddbea55a28616cc790b00aea5bc89b6649e56
    ↪ "
]
}
}
```

18.5.14 2.4.14 account_auth_platform

The account authorizes the platform.

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

The Owner key

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

account Authorizing accountUID or nickname

platform_owner platform owner accountUID or nickname

broadcast Whether to broadcast

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "account_auth_platform", ["250926091",
↪ "223331844", true]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params":[0, "account_auth_platform
↪", ["250926091", "223331844", true]], "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "ref_block_num": 26612,
    "ref_block_prefix": 1858930703,
    "expiration": "2018-07-05T02:34:57",
    "operations": [
      [
        23,
        {
          "fee": {
            "total": {
              "amount": 10000,
              "asset_id": 0
            },
            "options": {
              "from_csaf": {
                "amount": 10000,
                "asset_id": 0
              }
            }
          },
          "uid": 223331844,
          "platform": 250926091
        }
      ]
    ],
    "signatures": [
      ↪ "200ee643c33b074ad002bc6f4b477ff48dbee76bd3aa1c3c5b3a4c064b1f39581e61f93e957128c1c20737eafa2e09c56"
      ↪ "
    ]
  }
}
```

18.5.15 2.4.15 account_cancel_auth_platform

The account unauthorizes the platform.

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

The Owner key

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

account Authorizing accountUID of nickname

platform_owner platform owner accountUID or nickname

broadcast Whether to broadcast

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "account_cancel_auth_platform", ["250926091",
↪"223331844", true]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "account_cancel_auth_
↪platform", ["250926091", "223331844", true], "id": 1}' http://localhost:8091/rpc
```

Return results

```

{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "ref_block_num": 26736,
    "ref_block_prefix": 2510951750,
    "expiration": "2018-07-05T02:41:09",
    "operations": [
      [
        24,
        {
          "fee": {
            "total": {
              "amount": 10000,
              "asset_id": 0
            },
            "options": {
              "from_csaf": {
                "amount": 10000,
                "asset_id": 0
              }
            }
          },
          "uid": 223331844,
          "platform": 250926091
        }
      ]
    ],
    "signatures": [
      ↪ "2021240ab7c321fad009693c8a598363aa840fcde0f07cc54d72b8c3b7d6e116d4357127328faea805aff7d632fb0a09e2"
      ↪ ""
    ]
  }
}

```

18.5.16 2.4.16 create_asset

Creating assets

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

It needs the Active key of the applier.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

issuer Creator UID

symbol The symbol to create an asset

precision Precision (retains a few decimal places)

common Options, see the option parameter structure below

initial_supply The initial liquidity is an integer representation, ie: $\text{actual amount} = \text{initial_supply} / (10^{\text{precision}})$

broadcast Whether to broadcast

Option parameter structure

```
asset_options {
    // The maximum supply that the asset may have at any given time. This can be as_
    ↪big as GRAPHENE_MAX_SHARE_SUPPLY.
    // Special note: The maximum supply is the number of minimum unit of tokens.
    ↪For example, the maximum supply is 30,000 and the precision is set to 2 so the_
    ↪actual supply is 30000/(10^2)=300 Token. The smallest trading unit is 0.01.
    max_supply = GRAPHENE_MAX_SHARE_SUPPLY;
    // (reserved field, not used yet, must be 0)
    Market_fee_percent = 0;
    // (reserved field, not used yet, must be 0)
    max_market_fee = GRAPHENE_MAX_SHARE_SUPPLY;

    // The logo that the issuer has the right to update
    issuer_permissions = UIA_ASSET_ISSUER_PERMISSION_MASK;
    // current activity flag on this permission
    flags = 0;

    // This asset can be used by a group of whitelisted accounts. If whitelist_
    ↪authorities is not empty, only accounts in whitelist_authorities can hold, use or_
    ↪transfer assets.
    whitelist_authorities;
    // A set of blacklisted accounts cannot hold and use the assets.
    blacklist_authorities;

    // Define the assets that the asset may trade on the market (reserved fields,
    ↪not used yet)
    whitelist_markets;
    // Define assets that the asset must not trade on the market, and must not_
    ↪overlap whitelists (reserved fields, not used yet)
    blacklist_markets;

    // Data describing the meaning/purpose of the asset, the fee will be charged_
    ↪according to the size of the description.
    string description;
};
```

The issuer_permissions and flags fields are represented by integers. After the integer is converted to binary, each bit represents a permission or flag. Currently defined as follows:

```
enum asset_issuer_permission_flags
{
    white_list          = 0x02,    // whitelist, if enabled, the asset issuer can
    ↪control whether others can use the assets ("use" here includes transfers, etc.)
    override_authority = 0x04,    // Forced transfer, if enabled, the asset
    ↪issuer can force a transfer or take back the assets in another person's account
    transfer_restricted = 0x08,    // Restricted transfer, if enabled, the
    ↪originator or recipient of the transfer must be the issuer of the assets
    issue_asset        = 0x200,    // Issuing the assets, if enabled, the asset
    ↪issuer can add a certain amount of that assets to an account and increase the
    ↪current total circulation amount of the assets
    change_max_supply  = 0x400,    // modifying the circulation limit, if enabled,
    ↪the asset issuer can modify the circulation limit of the assets
};
```

Bits that are not currently in use must be 0 in both issuer_permissions and flags fields. In the flags field, if a bit is 1, it means that the corresponding parameter is enabled for the asset; if it is 0, it means that it is not enabled. In the issuer_permissions field, if a bit is 0, it means that the issuer of the asset can modify the corresponding parameter bit in the flags field; if it is 1, it means that it cannot be modified.

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "create_asset", ["250926091","TOTOTO", 4, {
    ↪"max_supply":300000,"market_fee_percent":0,"max_market_fee":0,"issuer_permissions
    ↪":4}, 200000, true]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params":[0, "create_asset", [
    ↪"250926091","TOTOTO", 4, {"max_supply":300000,"market_fee_percent":0,"max_market_fee
    ↪":0,"issuer_permissions":4}, 200000, true]], "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "ref_block_num": 57735,
  "ref_block_prefix": 998946957,
  "expiration": "2018-07-06T04:31:06",
  "operations": [[
    25, {
      "fee": {
        "total": {
          "amount": 50000000,
```

(continues on next page)

(continued from previous page)

```

        "asset_id": 0
    },
    "issuer": 250926091,
    "symbol": "TOTOTO",
    "precision": 4,
    "common_options": {
        "max_supply": 300000,
        "market_fee_percent": 0,
        "max_market_fee": 0,
        "issuer_permissions": 4,
        "flags": 0,
        "whitelist_authorities": [],
        "blacklist_authorities": [],
        "whitelist_markets": [],
        "blacklist_markets": [],
        "description": ""
    },
    "extensions": {
        "initial_supply": 200000
    }
}
]
],
"signatures": [
↪ "1f32dc80319019ff0c5eff27c9d1e15c7371fcc2825969b8e42a6429fb44c2ac3a75344b382e00c9f7f4af06600b0aff9
↪ "
]
}

```

18.5.17 2.4.17 update_asset

Update asset information

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

It needs the Active key of the asset owner.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

symbol asset symbol

new_issuer new asset owner

new_options New asset option (see common parameter structure in `create_asset`); enter null if without modification

broadcast Whether to broadcast

Precautions

This feature is only available to asset issuers.

The precision can only be modified if the current circulation amount is zero.

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "update_asset", ["WOWO", null, {"max_supply":
↪"20000000000"}, true]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params":[0, "update_asset", ["WOWO
↪", null, {"max_supply":"20000000000"}, true]]}, "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "ref_block_num": 12862,
    "ref_block_prefix": 653302827,
    "expiration": "2018-05-04T04:05:39",
    "operations": [
      [
        26,
        {
          "fee": {
            "total": {
              "amount": 50000000,
              "asset_id": 0
            }
          },
          "issuer": 223331844,
          "asset_to_update": 3,
          "new_options": {
            "max_supply": 2000000000,
            "market_fee_percent": 0,
            "max_market_fee": "10000000000000000",

```

(continues on next page)

(continued from previous page)

```
        "issuer_permissions": 79,
        "flags": 0,
        "whitelist_authorities": [],
        "blacklist_authorities": [],
        "whitelist_markets": [],
        "blacklist_markets": [],
        "description": ""
    }
}
],
"signatures": [
↪ "2030b2b084ab8e47bc2da5c863475776eac2cde1feba4cb3575eb7a7e86f96c9594a9778280c740098a5eec26f84b5799"
↪ "
]
}
}
```

18.5.18 2.4.18 enable_allowed_assets

The account actively enables or disables the whitelist of account-side assets.

The whitelist is deactivated by default, and when the whitelist is disabled, the account can send and receive any assets.

When the whitelist is enabled, the account can only send and receive assets in the list, and the whitelist can be updated using the `update_allowed_assets` command.

When the state is changed from deactivated to enabled, there is only “core assets” in the whitelist by default, which is YOYO.

Data is cleared when the whitelist is deactivated from the enabled state. To re-enable it, the required assets need to be re-added.

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

It needs the Active key of the asset owner.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

- account** AccountUID or nickname
- enable** Whether to enable true is to enable false is to disable
- broadcast** Whether to broadcast

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "enable_allowed_assets", ["250926091", false, ↵
↵true]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params":[0, "issue_asset", ↵
↵"250926091", false, true]], "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "ref_block_num": 56000,
  "ref_block_prefix": 3273656390,
  "expiration": "2018-07-06T03:04:21",
  "operations": [[
    34, {
      "fee": {
        "total": {
          "amount": 100000,
          "asset_id": 0
        },
        "options": {
          "from_csaf": {
            "amount": 100000,
            "asset_id": 0
          }
        }
      },
      "account": 250926091,
      "enable": false
    }
  ]],
  "signatures": [
```

(continues on next page)

(continued from previous page)

```
↔ "1f7e06ed28f6017e9a25ee1d165ec1aae330fb87abc9486c264c30f7de2e8d2369077509d61d96df016fbec55bbd5415"
↔ "
  ]
}
```

18.5.19 2.4.19 update_allowed_assets

Update the whitelist of account-side assets

Updates can only be made when the whitelist is open. Cannot remove YOYO.

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

It needs the Active key of the asset owner.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

account AccountUID or nickname

assets_to_add List of assets added to the whitelist (asset code or id)

assets_to_remove List of assets removed from the whitelist (asset code or id)

broadcast Whether to broadcast

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "issue_asset", ["250926091", "100000", "WOWO",
↪"memo", true]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params":[0, "issue_asset", [
↪"250926091", "100000", "WOWO", "memo", true]], "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "ref_block_num": 56441,
  "ref_block_prefix": 3725987382,
  "expiration": "2018-07-06T03:26:24",
  "operations": [[
    35, {
      "fee": {
        "total": {
          "amount": 200000,
          "asset_id": 0
        },
        "options": {
          "from_csaf": {
            "amount": 200000,
            "asset_id": 0
          }
        }
      },
      "account": 250926091,
      "assets_to_add": [
        3
      ],
      "assets_to_remove": []
    }
  ]],
  "signatures": [
    ↪"200f4362763eef9a5f0de6c461d2dcde77e665ae5ac2d462c89dd1f4ba22b35b7b21662e306d1e52c530815983f284d791",
    ↪",
    ↪"1f199078220bf0d6c2fcf28e0bd4fbf9afc1eb7db241997173a469d88d538d36c03c3299136cf42a030810c5adf250be9",
    ↪"
  ]
}
```

18.5.20 2.4.20 issue_asset

Assign the issued assets to an account

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

It needs the Active key of the asset owner.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

to_account Target account issued

amount amount

symbol asset symbol

memo memo

broadcast Whether to broadcast

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "issue_asset", ["250926091", "100000", "WOWO",
↪"memo", true]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params": [0, "issue_asset", [
↪"250926091", "100000", "WOWO", "memo", true]], "id": 1}' http://localhost:8091/rpc
```

Return results

```

{
  "ref_block_num": 55598,
  "ref_block_prefix": 1065296620,
  "expiration": "2018-05-03T09:05:39",
  "operations": [[
    27, {
      "fee": {
        "total": {
          "amount": 2008984,
          "asset_id": 0
        }
      },
      "issuer": 223331844,
      "asset_to_issue": {
        "amount": 1000000000,
        "asset_id": 3
      },
      "issue_to_account": 250926091,
      "memo": {
        "from": "YYW8EeMDaSmDg8zLXL272kcm4W7vUF4c4tBkfQpV7N79PBjgGjkdN",
        "to": "YYW7SpC4QLY1LRRxFQ2hbYHdAyQo88L8qnPJcDJkiRMugcnFGUGvo",
        "nonce": "3388232258004121975",
        "message": "fc712e427e57aae3d6819e3427b7f90a"
      }
    }
  ]],
  "signatures": [
    ↪ "205e29664c74590e1af73aa645374dc25f2900758e5213a3120fdf285d7940d5e66339311a97513cf78bd794916619f58"
    ↪ ""
  ]
}

```

18.5.21 2.4.21 reserve_asset

Destroy the specified number of specified assets in your account. After the operation is completed, the total amount of circulation of the asset type is correspondingly reduced.

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

It needs the Active key of the asset owner.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

from AccountUID or nickname

amount Amount

symbol Token typeasset code

broadcast Whether to broadcast

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "reserve_asset", ["250926091", "1000", "WOWO", ↵
↵true]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params":[0, "reserve_asset", [
↵"250926091", "1000", "WOWO", true]], "id": 1}' http://localhost:8091/rpc
```

Return results

```
{
  "ref_block_num": 56703,
  "ref_block_prefix": 1905534427,
  "expiration": "2018-07-06T03:39:30",
  "operations": [[
    28, {
      "fee": {
        "total": {
          "amount": 10000,
          "asset_id": 0
        },
        "options": {
          "from_csaf": {
            "amount": 10000,
            "asset_id": 0
          }
        }
      },
      "payer": 250926091,
```

(continues on next page)

(continued from previous page)

```

    "amount_to_reserve": {
      "amount": 10000000,
      "asset_id": 3
    }
  ],
  "signatures": [
    ↪ "1f6a2f416a525ae4754a98388cb4464f60f1b77659c5a742d8b76a80d581484ce5681c24aa51c7379acaa8247df49d962
    ↪ "
  ]
}

```

18.5.22 2.4.22 override_transfer

Forced transfer. An asset issuer forces an account to transfer a certain amount of assets to another account.

Only asset issuers can use this feature.

Remarks are encrypted with the issuer's Memo key, and the transferee can decrypt the remarks, and the transferrer cannot decrypt.

Supported format

JSON

Request method

WebSocket; JSON-RPC

Required private key authority

It needs the Active key and Memo key of the asset issuer.

Access authorization limit

Access level: normal interface

Frequency limit: True

Request parameters

from TransferrerUID or nickname

to TransfereeUID or nickname

amount Amount

symbol Token typeasset code

memo Memouse empty string "" if without a memo

broadcast Whether to broadcast

Precautions

null

Call sample and debug tools

WebSocket:

```
wscat -c ws://localhost:8091
{"id":1, "method":"call", "params":[0, "override_transfer", ["216494599", "250926091",
↪ 2, "SOSO", "memo", true]]}
```

JSON-RPC:

```
curl --data '{"jsonrpc": "2.0", "method": "call", "params":[0, "override_transfer", [
↪ "216494599", "250926091", 2, "SOSO", "memo", true]], "id": 1}' http://
↪ localhost:8091/rpc
```

Return results

```
{
  "ref_block_num": 61983,
  "ref_block_prefix": 2781522721,
  "expiration": "2018-07-06T08:03:30",
  "operations": [[
    30, {
      "fee": {
        "total": {
          "amount": 2008984,
          "asset_id": 0
        },
        "options": {
          "from_csaf": {
            "amount": 2008984,
            "asset_id": 0
          }
        }
      },
      "issuer": 223331844,
      "from": 216494599,
      "to": 250926091,
      "amount": {
        "amount": 20000,
        "asset_id": 283
      },
      "memo": {
        "from": "YYW5jiVsASqAyUibUb8awgDapFqfjQJo3SfvTEPJHxir8r3ggVtZU",
        "to": "YYW8Z8rtp7oEZFJi7Ar9HC4r4d15jXCVTQ2CAosjSnkSFdbc74GJ3",
        "nonce": "9585190871224727019",
        "message": "b0e7b1997c2142125c0369c4304a33c1"
      }
    }
  ]
}
```

(continues on next page)

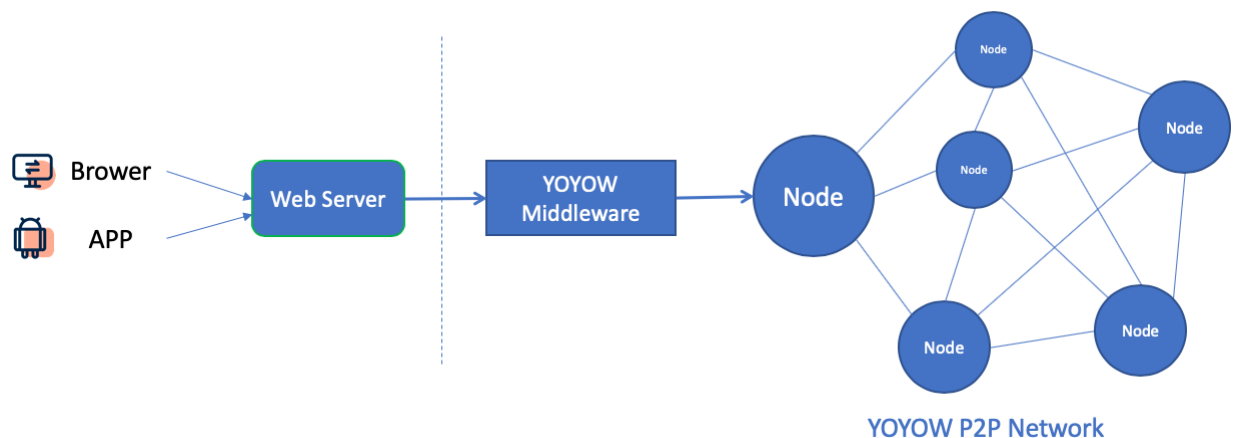
(continued from previous page)

```
    ]
  ],
  "signatures": [
    ↪ "20008db33ea71863fc7f4b97763a34e96e8b9d87714b1445d33c34a6b7bb3015c14d0a130c1ee74946c19b5e992c02249
    ↪ "
  ]
}
```

YOYOW Middleware InstructionNodejs

Using YOYOW middleware is the easiest way to integrate with the platform. It mainly provides three interfaces: account authorization, platform incentives and content chaining. You can use Docker one-click deployment to get the corresponding API, and easily interact with the YOYOW blockchain.

YOYOW middleware communicates with YOYOW network through the API interface of YOYOW node, which provides platform service providers with convenient access to data on the chain, ensuring that the traditional business codes can also reach the requirements of being on chain with only minimal changes. The specific diagram is as follows:



For the creation steps of the platform, please refer to: [Create a YOYOW platform account from scratch](#)

19.1 Deployment Start

19.1.1 Configuration File Description

The path to the configuration file is in the `conf/config.js` file in the code path. If you start it in docker mode, you can map the configuration file to the `/app/conf` directory in the container.

```
{
  // The api server address, the testnet public api address is as follows, for the
  ↳official network deployment, please change the address
  apiServer: "ws://47.52.155.181:10011",

  // The validity time of the security request, and the unit is "s". If the
  ↳requested content exceeds the validity period, it will return 1003 the request has
  ↳expired.
  secure_ageing: 60,

  // The platform security request verification key can be customized. For details,
  ↳see "Security Access".
  secure_key: "",

  // Platform owner active key
  active_key: "",

  // Platform owner secondary key
  secondary_key: "",

  // Platform owner memo key
  memo_key: "",

  // Platform id (yoyow id)
  platform_id: "",

  // Whether to use points for the operating fee
  use_csaf: true,

  // Whether the transfer is transferred to the balance, otherwise it is
  ↳transferred to tipping
  to_balance: false,

  // Wallet authorization page URL, testnet address is as follows, official network
  ↳address "https://wallet.yoyow.org/#/authorize-service"
  wallet_url: "http://demo.yoyow.org:8000/#/authorize-service",

  // The IP list that is allowed to access forcing the specific IP address to be
  ↳specified. "*" or "0.0.0.0" is not supported at this time.
  allow_ip: ["localhost", "127.0.0.1"]
}
```

Note:

1. In the general use scenario, the middleware value needs to use the secondary key and the memo key at most, and just the secondary key and the memo key can satisfy most of the requirements. Do not write the active key into the configuration file unless you are sure you need to use the active key.
2. The middleware uses the restriction IP (`allow_ip`) and encryption request (`secure_key`) to ensure security. However, it is still strongly recommended that the intranet be deployed and isolated, and the security of the private key is quite important.
3. It is recommended to use the point deduction for the operation fee. If the deduction fails, it will directly report the error and will not automatically deduct the tipping as the fee.

19.1.2 Docker One-Click Deployment

```
docker run -itd --name yoyow-middleware -v <Local configuration file path>:/app/conf -
↳p 3001:3001 yoyoworg/yoyow-middleware
```

19.1.3 Manual Deployment

1. clone source code `git clone git@github.com:yoyow-org/yoyow-node-sdk.git`
2. Modify middleware configuration modify the file `yoyow-node-sdk/middleware/conf/config.js` with reference to the configuration file description ().
3. The node library required to install the middleware service; go to the `~/yoyow-node-sdk/middleware/` directory and find `npm install`.
4. Start middleware service `npm start`

```
Api status: open
Login done
chain_id: 3505e367fe6cde243f2a1c39bd8e58557e23271dd6cbf4b29a8dc8c44c9af8f
network_name: Test
Api connected.....
synced , chainstore ready
HTTP opened port: 3000
```

Normal start as shown below

19.2 Interface Descriptions

19.2.1 Request Documentation and Examples

1. Basic Query Related Interface

1.1. Get the Specified Account Information

getAccount

Request Type GET

Request Parameters

```
{Number} uid - account id
```

Request Example

```
localhost:3000/api/v1/getAccount?uid=25638
```

Return Results

```
{
  code: results,
  message: return message,
  data: { // user information
    uid: account uid
```

(continues on next page)

(continued from previous page)

```

name: account name
owner: owner authority
active: active authority
secondary: secondary authority
memo_key: memo key public key
reg_info: registration information
can_post: can post or not
can_reply: can reply to posts or not
can_rate: can comment or not
is_full_member: is full member or not
is_registrar: is registrar or not
is_admin: is administrator or not
statistics: { //user YOYO asset details
  obj_id: asset object id
  core_balance: balance
  prepaid: tipping
  csaf: bonus points
  total_witness_pledge: witnesses' total collateral (collateral amount of
↳creating witnesses by users)
  total_committee_member_pledge: committee total collateralcollateral amount of
↳creating committee by users
  total_platform_pledge: platform total collateralcollateral amount of creating
↳platforms by users
  releasing_witness_pledge: witness collateral to be returned
  releasing_committee_member_pledge: committee collateral to be returned
  releasing_platform_pledge: platform collateral to be returned
}
assets: [ //total assets held by users
  {
    amount: asset amount,
    asset_id: asset id,
    precision: asset precision,
    symbol: asset symbol,
    description: asset description"
  }
  ...
]
}
}

```

1.2. Get Recent Activity Records for a Given account

getHistory Request TypeGET

Request Parameters

```

{Number} uid - account id
{Number} op_type - query op type '0' is transfer op, default is null, i.e. query all
↳op types
{Number} start - query start number. When it is 0, the query starts from the latest
↳record. The default is 0.
{Number} limit - the length of the querythe maximum cannot exceed 100, the default is
↳10.

```

Request Example


```
localhost:3000/api/v1/getHistory?uid=25638&start=1220&limit=30&op_type=0
```

Return Results

```
{
  code: operation results,
  message: return message,
  data: [] array of history objects
}
```

1.3. Verify Whether Block is Unreturnable

confirmBlock

Request TypeGET

Request Parameters

```
{Number} block_num - Verified block number
```

Request Example

```
localhost:3000/api/v1/confirmBlock?block_num=4303231
```

Return Results

```
{
  code: operation results,
  message: return message,
  data: is the block returnable or not
}
```

1.4. Get Specified Asset Information

getAsset

Request TypeGET

Request Parameters

```
{String | Number} search - asset symbolcapitalor asset id
```

Request Example

```
http://localhost:3001/api/v1/getAsset?search=YOYOW
```

Return Results

```
{
  code: operation results,
  message: return message,
  data: {
    "id":"1.3.0", - asset object id
    "asset_id":0, - asset id
    "symbol":"YOYO", - asset symbol
  }
}
```

(continues on next page)

(continued from previous page)

```

    "precision":5, - asset precision
    "issuer":1264, - asset issuer uid
    "options":{
      "max_supply":"2000000000000000", - circulation limit
      "market_fee_percent":0, - transaction fee percentage
      "max_market_fee":"1000000000000000", - transaction fee maximum
      "issuer_permissions":0, - asset availability
      "flags":0, - asset authority
      "whitelist_authorities":[], - asset whitelist administrator list
      "blacklist_authorities":[], - asset blacklist administrator list
      "whitelist_markets":[], - trading pair whitelist
      "blacklist_markets":[], - trading pair blacklist
      "description":"" - asset description
    },
    "dynamic_asset_data_id":"2.2.0", - asset dynamic object id
    "dynamic_asset_data":{
      "id":"2.2.0", - asset dynamic object id
      "asset_id":0,
      "current_supply":"107384564466939", - current supply of assets
      "accumulated_fees":0
    },
    "current_supply":"107384564466939", - current supply of assets
    "accumulated_fees":0
  }
}

```

1.5. Get Specified Platform Information

getPlatformById

Request TypeGET

Request Parameters {Number} uid - platform owner account uid

Request Example

```
http://localhost:3001/api/v1/getPlatformById?uid=217895094
```

Return Results

```

{
  "id": "1.6.0", - platform object id
  "owner": 217895094, - platform owner account uid
  "name": "test-yoyow", - platform name
  "sequence": 1,
  "is_valid": true, - is valid or not
  "total_votes": 0, - platform total votes
  "url": "http://demo.yoyow.org/", - platform url address
  "pledge": 1000000000, - platform collateralYOYO
  "pledge_last_update": "2018-02-10T01:03:57", - platform collateral last update_
↪time
  "average_pledge": 176601774, - platform average collateral
  "average_pledge_last_update": "2018-02-11T06:49:12", - platform average_
↪collateral last update time
  "average_pledge_next_update_block": 4562164, - platform average collateral next_
↪update block number
}

```

(continues on next page)

(continued from previous page)

```

"extra_data": "{}", - platform extra data
"create_time": "2018-02-10T01:03:57", - platform creating time
"last_update_time": "2018-02-11T06:49:12" - platform latest update date
}

```

2. Platform Incentives Related Interface

2.1. Transfer to Specified User

transfer requiring security verification request

Request Type POST

Request Parameters

```

{Object} cipher - Request ciphertext object the format is as follows
{
  ct, - ciphertext hexadecimal
  iv, - vector hexadecimal
  s - salt hexadecimal
}

```

Request Object Structure:

```

{Number} uid - specified user id
{Number} amount - transfer amount
{Number} asset_id - asset id
{string} memo - memo
{Number} time - operation time

```

Request Example refer to security request verification

```
localhost:3000/api/v1/transfer
```

Return Results

```

{
  code: operation results,
  message: return message,
  data: {
    block_num: operation block number
    txid: operation id
  }
}

```

2.2. Get Transfer QR Code Text

getQRReceive this QR code can be used for YOYOW APP

Request Type GET

Request Parameters

```
{Number} amount - receipt amount If the receipt note is not filled, the user can_
↳enter it in the APP

{String} memo - receipt memo If the receipt amount is not filled, the user can enter_
↳it in the APP

{String | Number} asset - transfer asset symbol or asset IDdefault is YOYO asset
```

Request Example

```
http://localhost:3001/api/v1/getQRReceive?amount=98&memo=new transfer&asset_id=0
```

Return Results

```
{
  code: operation results,
  message: return message,
  data: receipt QR code string
}
```

2.3. Modify (whitelist only) Authorized User Asset Whitelist

updateAllowedAssets (requiring security verification request)

If the user has enabled asset whitelisting, then UIA (user-issued assets) needs to be added to the user's asset whitelist before transactions such as transfers can be made.

Request TypePOST

Request Parameters

```
{Object} cipher - Request object ciphertext object

{
  ct, - cipertext hexadecimal
  iv, - vector hexadecimal
  s - salt hexadecimal
}
```

Request Object Structure:

```
{Number} uid - target account id

{Number} asset_id - asset id
```

Request Examplerefer to Security Request Verification

```
localhost:3000/api/v1/updateAllowedAssets
```

Return Results

```
{
  code: operation results,
  message: return message,
  data: {
    block_num: operation block number
  }
}
```

(continues on next page)

(continued from previous page)

```

    txid: operation id
  }
}

```

3. Content Chaining Related Interface

3.1. Send Post (requiring for security verification request)

Request Type POST

Request Parameters

```

{Object} cipher - Request object ciphertext object
{
  ct, - ciphertext hexadecimal
  iv, - vecotr hexadecimal
  s - salt hexadecimal
}

```

Request Object Structure:

```

{Number} platform - platform account
{Number} poster - poster account
{Number} post_pid - post number
{String} title - post title
{String} body - post content
{String} extra_data - post extra data
{String} hash_value - hash value, the defalut value will be the sha256 value of the_
↳body content if not provided.
{Number} origin_platform - original post platform account default null
{Number} origin_poster - original post poster account default null
{Number} origin_post_pid - original post number default null
{Number} time - operation time

```

Request Example: refer to Security Request Verification

```
localhost:3000/api/v1/post
```

Return Results

```

{
  code: operation results,
  message: return message,
  data: {
    block_num: operation block number
    txid: operation id
  }
}

```

3.2. Update Post

postUpdate (requiring security verification request)

Request TypePOST

Request Parameters

```
{Object} cipher - Request object ciphertext object
{
  ct, - ciphertext hexadecimal
  iv, - vector hexadecimal
  s - salt hexadecimal
}
```

Request Object Structure:

```
{Number} platform - platform account
{Number} poster - poster account
{Number} post_pid - post number
{String} title - post title
{String} body - post content
{String} extra_data - post extra data
{Number} time - operation time
```

Note: When modifying the post operation, at least one of title, body and extra_data must appear, and the content of the same field as the original text is different.

Request Example: refer to Security Request Verification

```
localhost:3000/api/v1/postUpdate
```

Return Results

```
{
  code: operation results,
  message: return message,
  data: {
    block_num: operation block number
    txid: operation id
  }
}
```

3.3. Get Post

getPost

Request TypeGET

Request Parameters

```
{Number} platform - platform account
{Number} poster -poster account
{Number} post_pid - post number
```

Request Example

```
http://localhost:3001/api/v1/getPost?platform=217895094&poster=210425155&post_pid=3
```

Return Results

```
{
  code: operation results,
  message: return message,
  data: {
    "id": "1.7.12", - post objectId
    "platform": 217895094, - platform account
    "poster": 210425155, - poster account
    "post_pid": 5, - post number
    "hash_value": "bb76a28981710f513479fa0d11fee154795943146f364da699836fb1f375875f
↪", - post body hash value
    "extra_data": "{}", - extra info
    "title": "test title in js for update", - post title
    "body": "test boyd in js for update", - post content
    "create_time": "2018-03-12T10:22:03", - post creating time
    "last_update_time": "2018-03-12T10:23:24", - post latest update time
    "origin_platform", - original post platform account Only exist for forwarding ↪
↪ a post when posting
    "origin_poster", - original post poster account Only exist for forwarding a ↪
↪ post when posting
    "origin_post_pid" - original post posting number Only exist for forwarding a ↪
↪ post when posting
  }
}
```

3.4. Get Post List

getPostList

Request Type GET

Request Parameters

```
{Number} platform - platform account
{Number} poster - poster account default null query all posts on the platform when it ↪
↪ is null
{Number} limit - load number default 20
{String} start - starting time 'yyyy-MM-ddThh:mm:ss' ISOString When the next page is ↪
↪ loaded, the last create_time of the currently loaded data is passed in. If it is ↪
↪ not passed, it is loaded from the beginning.
```

Request Example

```
http://localhost:3001/api/v1/getPostList?platform=217895094&poster=210425155&limit=2&
↪ start=2018-03-12T09:35:36
```

Return Results

```
{
  code: operation results,
  message: return message,
```

(continues on next page)

(continued from previous page)

```
data: [post objectRefer to the returned data structure of getting a single post]
}
```

4. About Auth

4.1. Signature Platform

sign

Request TypeGET

Request Parametersnull

Request Example

```
localhost:3000/auth/sign
```

Return Results

```
{
  code: operation results,
  message: return message,
  data: {
    sign: signature results,
    time: operation time millisecond value,
    platform: platform owner id,
    url: wallet authorization url
  }
}
```

4.2 Signature Verification

verify

Request TypeGET

Request Parameters

```
{Number} yoyow - account id
{Number} time - operation time millisecond value
{String} sign - signature results
```

Request Example

```
localhost:3000/auth/verify?
↪sign=20724e65c0d763a0cc99436ab79b95c02fbb3f352e3f9f749716b6dac84c1dc27e5e34ff8f0499ba7d94f1d14098c
↪time=1517534429858&yoyow=217895094
```

Return Results

```
{
  code: operation results,
  message: return message,
  data: {
```

(continues on next page)

(continued from previous page)

```

verify: is the signature successful or not,
name: signed YOYOW user name
}
}

```

4.3 Signature Platform Returned QR Code

signQR

Request Type GET

Request Parameters

```

{String} state - The extra data will be sent to the platform together with the user_
↳signature information when the platform login interface is invoked. It is used when_
↳the platform login interface needs a customized parameter. If there is no such_
↳requirement, it may not be transmitted.

```

Request Example

```
localhost:3000/auth/signQR?state=platformCustomParams
```

Return Results

```

{
  code: operation results,
  message: return message,
  data: QR code picture base64 string
}

```

4.4 Platform Extra Data Protocol Descriptions

platform attributes extra_data extra data JSON object format string

```

{
  "login": "http://example/login" //Platform QR code scanning login request interface
  "description": "platform description" //platform description
  "image": "http://example.image.jpg" //platform imageplatform image displayed in_
↳yoyow app 1.1
  "h5url": "http://exampleH5.com" //Platform h5 address, used to adjust the h5 page_
↳without the app jumping
  "packagename": "com.example.app" //Platform android jump
  "urlscheme": "example://" //Platform ios jump
}

```

4.5 Platform Login by Scanning QR Code

When the wallet App scans QR code and it will access and post signature object to “login” url in extra data.

```

{
  {Number} yoyow - Current operating user account id
  {String} time - Signature timestamp string
}

```

(continues on next page)

(continued from previous page)

```
{String} sign - Signature string
{String} state - Custom information passed in when the platform is signing (refer_
↳to About Auth 4.3 - signQR)
}
```

the interface provided by the platform must return the following information

```
{
{Number} code - operation result 0 means passing. Any non-zero condition is_
↳considered an error
{String} message - operation result description
}
```

19.2.2 Request for Returning Error Code Status Description

```
1001 invalid signature type
1002 invalid signature time
1003 request has expired
1004 invalid operation time
1005 invalid operation signature
1006 account information does not match the chain (usually after the private key is_
↳restored, using the local data of other computers or the old backup file for_
↳authorization operation)
1007 Unauthorized platform
2000 api underlying exception
2001 account does not exist
2002 invalid account
2003 invalid transfer amount
2004 tipping and points are insufficient for paying fees
2005 insufficient tipping
2006 invalid asset symbol or id
3001 Post ID must be the previous post ID +1 of the issuer of the platform (platform_
↳post management id)
```

19.2.3 Security Request Verification

Operations related to financial security, such as transfer, posting, and other write operations, will be verified for their effectiveness in the middleware service. The information of such requests needs to be converted into ciphertext by

encryption and then sent to the middleware service. The encryption method uses symmetric encryption AES, and the key is `secure_key` in the configuration file.

Encryption example (crypto-js version of javascript, other languages use similar AES encryption)

Default mode CBC , padding scheme Pkcs7

For example transfer operation

```

let key = 'customkey123456'; // This key is the same as the secure_key in the_
↳config in the middleware.

let sendObj = {
  "uid": 9638251,
  "amount": 100,
  "asset_id": 0,
  "memo": "hello yoyow",
  "time": Date.now() //time field
  The operation time takes the current time millisecond value.
  Encryption must have this field for verifying the operation time
}

let cipher = CryptoJS.AES.encrypt(JSON.stringify(sendObj), key);

$.ajax({
  url: 'localhost:3000/api/v1/transfer',
  type: 'POST',
  data: {
    ct: cipher.ciphertext.toString(CryptoJS.enc.Hex),
    iv: cipher.iv.toString(),
    s: cipher.salt.toString()
  },
  success: function(res){
    // do something ...
  }
})

```

PHP encryption

```

function cryptoJsAesEncrypt($passphrase, $value) {
  $salt = openssl_random_pseudo_bytes(8);
  $salted = '';
  $dx = '';
  while (strlen($salted) < 48) {
    $dx = md5($dx.$passphrase.$salt, true);
    $salted .= $dx;
  }
  $key = substr($salted, 0, 32);
  $iv = substr($salted, 32, 16);
  $encrypted_data = openssl_encrypt($value, 'aes-256-cbc', $key, true, $iv);
  $data = array("ct" => bin2hex($encrypted_data), "iv" => bin2hex($iv), "s" =>
↳bin2hex($salt));
  return json_encode($data);
}

```

For other operations that require secure request verification, change `sendObj` according to the documentation

CHAPTER 20

Descriptions of YOYOW Parameter Table and Fees

Please refer to the YOYOW Browser:

- [Fees](#)
- [Chain Parameters](#)

Instructions for Exchanges

We here describe how to integrate your exchange with the YOYOW blockchain step-by-step.

21.1 Preparation

21.1.1 Hardware

The recommended hardware by now is a VPS with 2GB of RAM and 20GB HDD. Single core is OK.

Supported platforms:

- Ubuntu 16.04 LTS 64 bit
- Windows Servers 64 bit

We'll take Ubuntu as an example in this document.

21.1.2 Services

Disable the default time-syncing daemon (timedated) and install NTPD:

```
sudo timedatectl set-ntp false
sudo apt-get -y install ntp
```

21.1.3 Account

Register an account in <https://wallet.yoyow.org>. Instruction is [here](#).

Get the private keys of your account. Instruction is [here](#). We'll need 3 keys for integration:

- Active key: for transferring funds out
- Secondary key: for collecting points to pay fees

- Memo key: for encrypting/decrypting memos

21.1.4 Business Logic

An exchange uses an account (or more) for processing deposits/withdrawals.

- Deposits: assign a unique identifier to every customer, such as user ID or salted hash of user ID. Every customer transfer funds to the exchange's account with different identifiers as MEMO so the exchange can know a deposit is from which customer.
- Withdrawals: transfer funds from the exchange's account to an account the customer requested, with an optional memo that the customer may have requested. An optional memo is needed, because some customers may want to withdraw funds to another exchange directly.

21.2 Installation

Please download the "Latest Version" of the executable files ([here](#)). After downloading, extract the files. For example,

```
https://github.com/yoyow-org/yoyow-core/releases/download/v0.2.1-180313/yoyow-node-v0.
↪2.1-ubuntu-20180313.tgz
https://github.com/yoyow-org/yoyow-core/releases/download/v0.2.1-180313/yoyow-client-
↪v0.2.1-ubuntu-20180313.tgz
tar xzf yoyow-node-v0.2.1-ubuntu-20180313.tgz
tar xzf yoyow-client-v0.2.1-ubuntu-20180313.tgz
```

Notes: please remember to replace the latest version of node and client program.

21.3 Starting YOYOW Node

The node need to be always running. One way to achieve this is to run it with screen. The installation is as follow:

```
sudo apt-get -y install screen
```

Start yoyow_node with screen:

```
screen -S yoyow_node
./yoyow_node --rpc-endpoint 127.0.0.1:8090
```

Note:

1. Start the node with `--rpc-endpoint` so we can interact with it via RPC call. In the example the node will listen on address `127.0.0.1` and port `8090`.
2. The following parameters indicate how many history records are kept for each account. The default value is `1000`. For exchanges, if there are more recharge and withdrawal records, consider setting a larger value, e.g.:

```
max-ops-per-account = 1000
```

modify to

```
max-ops-per-account = 1000000
```

It will retain one million data. Earlier data is deleted from memory and cannot be queried quickly (but still recorded on the chain).

1. The following two parameters will greatly reduce the memory required for the operation. The principle is not to save the historical data index that is not related to the exchange account.

```
track-account = [25638]
partial-operations = true
```

Please replace “25638” with the account ID you need. Note: The default track-account in config.ini has a “#” symbol and needs to be deleted.

If you need to monitor multiple accounts, use the following configuration:

```
track-account = [25638,25997]
partial-operations = true
```

The node then will download blocks from the P2P network. When it’s done (in sync), in the console there will be new messages showing every 3 seconds, like these:

```
789216ms th_a      application.cpp:574      handle_block      ] Got block:
↪#355797 00056dd54878c05849e2dcd731c9ee364398b0d2 time: 2017-09-18T19:13:09 latency:
↪216 ms from: 27662/init8 irreversible: 355787 (-10)
792527ms th_a      application.cpp:574      handle_block      ] Got block:
↪#355798 00056dd613f39f1ad6c0c3fdb00a56c60f44ab1c time: 2017-09-18T19:13:12 latency:
↪526 ms from: 499381505/yoyo499381505 irreversible: 355788 (-10)
```

If need to terminate the node, press Ctrl+C, or send SIGINT or SIGTERM signal to the process, then wait for a while, the node will stop by itself.

21.4 Starting YOYOW Client

21.4.1 First Run

Start another screen session, run `yoyow_client` inside:

```
screen -S yoyow_client
./yoyow_client -s ws://127.0.0.1:8090/ -H 127.0.0.1:8091
```

Note:

1. Use `-s` option to connect the client to the node.
2. Use `-H` option to start a HTTP-RPC service so we can interact with the client from another process, E.G. a script to process funds deposit/withdrawal.
3. The node won’t start listening on the RPC port until finished replaying, so please be patient in this case.
4. You can connect multiple clients to one node, but don’t use the same `-H` option.

For the first time when running `yoyow_client`, after connected to the node, it will show:

```
Please use the set_password method to initialize a new wallet before continuing
new >>>
```

So we need to set a password as shown below, and it will be used to create and encrypt a wallet file:

```
new >>> set_password my-password
set_password my-password
```

(continues on next page)

(continued from previous page)

```
null
locked >>>
```

When there is already a wallet file, if we run `yoyow_client`, it will show `locked >>>` as well.

Then we need to unlock the wallet:

```
locked >>> unlock my-password
unlock my-password
null
unlocked >>>
```

Now we import the 3 private keys into the client, they will be encrypted then saved in the wallet file. The syntax is:

```
import_key [account_ID] [wif_private_key]
```

Assume your account ID is 123456789, and you have the Active Key, Secondary Key and Memo Key, we need to execute `import_key` command 3 times, with one time for one key. For example:

```
unlocked >>> import_key 123456789 5Hqwx3xXMYZ55Pko9nzw34234234nXHcGfNQjNEL23424w7Py
import_key 123456789 5Hqwx3xXMYZ55Pko9nzw34234234nXHcGfNQjNEL23424w7Py
2993104ms th_a      wallet.cpp:820      save_wallet_file      ] saving_
↪wallet to file wallet.json
true
unlocked >>> import_key 123456789 5JKoYzQ4sYZoDYwreyrfsd32466MsCFNoxRE23nExaRi6SY3
import_key 123456789 5JKoYzQ4sYZoDYwreyrfsd32466MsCFNoxRE23nExaRi6SY3
2993104ms th_a      wallet.cpp:820      save_wallet_file      ] saving_
↪wallet to file wallet.json
true
unlocked >>> import_key 123456789 5HttjgBSb45368989etfhsserVtt69cWcExteq6RktpAYXNTT
import_key 123456789 5HttjgBSb45368989etfhsserVtt69cWcExteq6RktpAYXNTT
2993104ms th_a      wallet.cpp:820      save_wallet_file      ] saving_
↪wallet to file wallet.json
true
unlocked >>>
```

21.4.2 Command: `info`

We can check network status with `info` command:

```
unlocked >>> info
info
{
  "head_block_num": 377867,
  "head_block_id": "0005c40b41f6d79d762b1ff81c7affc7ae82a894",
  "head_block_time": "2017-09-18T19:37:39",
  "head_block_age": "0 second old",
  "last_irreversible_block_num": 377857,
  "chain_id": "3505e367fe6cde243f2a1c39bd8e58557e23271dd6cbf4b29a8dc8c44c9af8fe",
  "participation": "100.0000000000000000",
  "active_witnesses": [[
    ...
  ]
}
```

21.4.3 Command: `get_block`

We can get the details of a specified block with `get_block` command. The syntax is:

```
get_block [block_number]
```

For example:

```
unlocked >>> get_block 1
```

21.4.4 Command: `get_full_account`

We can check the account info with `get_full_account` command:

```
unlocked >>> get_full_account 123456789
get_full_account 123456789
{
  "account": {
    "uid": 123456789,
    ...
  },
  "statistics": {
    "owner": 123456789,
    "total_ops": 30220,
    "prepaid": 0,
    "csaf": 37424828,
    "core_balance": "44672014515",
    "core_leased_in": 0,
    ...
  }
}
```

Note: the “statistics” data is useful for integration.

- “csaf” means “points”, which are generated by holding YOYO and will be used to pay transaction fees. Balances have 5 decimal digits in YOYOW, and the currency is YOYO, so "csaf": 37424828 means 374.24828 YOYO.
- “core_balance” is balance of the account. "core_balance": "44672014515" means 446,720.14515 YOYO.
- Please be aware that numbers will be surrounded with quotation marks when bigger than 2^{32} , as shown above for `core_balance` but not for `csaf`.

21.4.5 Command: `transfer`

We can use the `transfer` command to transfer funds. The syntax is:

```
transfer [from] [to] [amount] YOYO [memo] [broadcast]
```

For example:

```
unlocked >>> transfer 123456789 987654321 1.2345 YOYO "thisismemo" true
```

Note:

- amount can only have at most 5 decimal digits.

- If setting `broadcast` to `true`, the signed transaction will be broadcast to the P2P network. If using `false` for testing, it will not be broadcast.

21.4.6 Command: `get_transaction_id`

We can use the `get_transaction_id` command to get the hash of a transaction. The syntax is:

```
get_transaction_id [transaction_in_json]
```

This command is useful for integration.

21.4.7 Command: `get_relative_account_history`

We can use the `get_relative_account_history` command to check our transaction history. The syntax is:

```
get_relative_account_history [account] [operation_type] [start] [limit] [end]
```

For example:

```
unlocked >>> get_relative_account_history 123456789 null 1 10 10
unlocked >>> get_relative_account_history 123456789 0 11 10 20
```

Note:

- For `operation_type`, use `null` to get all operations, use `0` to get transfer only.
- For `end`, use `0` to get the most recent records.
- Result will be in range of `[start, end]`; if `limit` is smaller than the number of records in `[start, end]`, the latest records will be returned.
- Result is sorted in “latest first” order.

21.4.8 Command: `collect_csaf`

We can use the `collect_csaf` command to collect points which will be needed to pay transaction fees. The syntax is:

```
collect_csaf [from_account] [to_account] [amount] YOYO [broadcast]
```

For example:

```
unlocked >>> collect_csaf 123456789 123456789 10 YOYO true
```

Note:

- If you have some YOYO in your account, it will accumulate points as time goes by. The accumulation speed has a linear relationship with account balance. Usually for exchanges the accumulated points should be enough to pay transaction fees.
- Points need to be collected through this command before can be used to pay transaction fees.
- Although assets in the account (balance or tipping) can be used to pay transaction fees as well in the back end, current implementation of `yoyow_client` will only try to pay fees with points (except raw-transaction signing). If you don't have enough points in the account, most commands will fail. So it's important to keep a certain amount of points in the account.

21.4.9 How to Close

Press Ctrl+D if the client is running in Ubuntu.

21.5 Accessing the Client via HTTP-RPC

When HTTP-RPC is enabled, we can access the client via HTTP-RPC call. All commands are usable. For example:

```
curl -d '{"jsonrpc": "2.0", "method": "info", "params": [], "id": 1}' http://127.0.0.1:8091/rpc
curl -d '{"jsonrpc": "2.0", "method": "transfer", "params": [123456789,123456789,"1", "YOYO",null,true], "id": 1}' http://127.0.0.1:8091/rpc
curl -d '{"jsonrpc": "2.0", "method": "get_relative_account_history", "params": [123456789,0,1,10,10], "id": 1}' http://127.0.0.1:8091/rpc
```

Note:

- Use `http` as protocol
- Request `/rpc` but not `/`
- Not like the interactive CLI, the results of HTTP-RPC calls are formatted in json
- In the request, amounts have 5 decimal digits; in the response, amounts have no decimal point. Instead, amounts are multiplied by 10^5 .
- When entering the amount, it is recommended to use a string for the amount. It is recommended to use the normal decimal format, such as “12345.6789”. Problems may occur if using the scientific notation format such as 1.23456789E4.

21.6 Processing Deposits

21.6.1 Checking Node Status

Get the `last_irreversible_block_num` data with the “`info`” command (via HTTP-RPC). Only blocks earlier than this block is reliable.

21.6.2 Checking Account History

1. Firstly, Use `get_relative_account_history` command/API to get the latest sequence number:

```
curl -d '{"jsonrpc": "2.0", "method": "get_relative_account_history", "params": [123456789,0,0,1,0], "id": 1}' http://127.0.0.1:8091/rpc
```

If the result, `response["result"]` is empty, it means we have no deposit at all. If it's not empty, we get `response["result"][0]["sequence"]` as `maximum_sequence`.

If `maximum_sequence` is bigger than the last sequence we've saved, it means there are new records to be processed.

1. Use `get_relative_account_history` command/API to check for new records. For example, if last recorded sequence is 100, and `maximum_sequence` is 200, we can check from 101, for at most 100 records, to 200. Refer to the following commands:

```
curl -d '{"jsonrpc": "2.0", "method": "get_relative_account_history", "params": [123456789,0,101,100,200], "id": 1}' http://127.0.0.1:8091/rpc
```

In the returned result, `result=response["result"]`, so `result` is an array. If the array is empty, it means no new deposit. If it's not empty, the N'th record `result[N]` should be like:

```
{
  "memo": "alb2c3d4",
  "description": "Transfer 100 YOYO from 204501630 to 123456789 -- Memo: alb2c3d4
  (Fee: 0.20898 YOYO)",
  "sequence": 101,
  "op": {
    "op": [
      0,
      {
        "fee": {
          "total": {
            "amount": 20898,
            "asset_id": 0
          },
          "options": {
            "from_balance": {
              "amount": 20898,
              "asset_id": 0
            }
          }
        },
        "from": 204501630,
        "to": 123456789,
        "amount": {
          "amount": 10000000,
          "asset_id": 0
        },
        "memo": {
          "from": "YYW6U528P71X6V87765245356aPBPPDwwRp7urUiXYtFLHmrXRsn3u",
          "to": "YYW5eA89yqwerhdfghrjtr3452376trtyU6LD7a1kmvwYa5h51rDxr",
          "nonce": "3457645755345345",
          "message": "0938457345937abcdef3098945"
        },
        "extensions": {
          "from_balance": {
            "amount": 10000000,
            "asset_id": 0
          },
          "to_balance": {
            "amount": 10000000,
            "asset_id": 0
          }
        }
      }
    ],
    "result": [
      0,
      {
      }
    ],
    "block_timestamp": "2017-09-08T11:14:15",
```

(continues on next page)

(continued from previous page)

```

    "block_num": 223355,
    "trx_in_block": 0,
    "op_in_trx": 0,
    "virtual_op": 8795
  }
},

```

- Get `result[N]["op"]["block_num"]`, if it's smaller than `last_irreversible_block_num`, then it's reliable, and need to be processed.
- Get `result[N]["op"]["op"][0]`, if it's 0, then it's a transfer. (Although it should always be 0 if we request transfers only)
- Get `result[N]["op"]["op"][1]["to"]`, if it's the same as our account ID, then this transfer is a deposit.
- Check if `result[N]["op"]["op"][1]["amount"]["asset_id"]` is 0. If yes, it means it's YOYO asset.
- Get `result[N]["op"]["op"][1]["amount"]["amount"]`, it's the amount. Remember to add the decimal point (5 digits).
- Get `result[N]["memo"]`, it is the memo info of the transfer records. It should have been decrypted already, and it can be an identifier to the customer.
- Save `result[N]["sequence"]` as the latest processed sequence number for use in the next loop.
- Save `result[N]["op"]["trx_in_block"]` for later use.
- Use `get_block` command/API to get this transfer's transaction ID/Hash (change the parameter to the `block_num`):

```

curl -d '{"jsonrpc": "2.0", "method": "get_block", "params": [160000], "id": 1}'_
↪http://127.0.0.1:8091/rpc

```

From the response `new_response`, save `new_response["result"]["transaction_ids"][trx_in_block]` as the transaction ID/hash of this deposit for future use.

Note:

- To be able to decrypt memo, the client need to be unlocked, and the memo private key need to be in the wallet.

21.7 Process Withdrawal Requests

21.7.1 Check Node Status

To be safe, we can only process withdrawal requests when node status is normal. Check with the `info` command/API.

- `head_block_time` should be no more than 15 seconds old
- `participation` should be more than 80, which means 80% of block producers are online and at normal statuses.

21.7.2 Checking Account Balance and Points

Check with the `get_full_account` command/API.

If points are not enough for paying transaction fee, collect more points with `collect_csaf` command/API.

21.7.3 Sending Funds

1. Use the `transfer` command/API to send out funds.

Note: pay attention to the decimal digits.

Save the returned json for future use.

1. Get the transaction ID/hash with `get_transaction_id` command/API, save it for future use.

21.7.4 Re-check / Follow-up

Similar to the deposit processing steps, when found an outgoing transfer, save the transaction ID/hash, block number and etc. for future use.

21.7.5 Trouble Shooting

Every transaction has an `expiration` field. If the transaction hasn't been included in any block for some reason, and the timestamp of block whose number is `last_irreversible_block_num` is later than the `expiration` field of the transaction, the transaction won't be included in current chain, so it's safe to try to transfer again.

21.8 Sample Code

- Ruby: <https://github.com/yoyow-org/yoyow-core/blob/master/scripts/exchange.rb>

Introduction to Platform Integration

YOYOW is a media public chain based on blockchain technology. Its goal is to build a network that utilizes blockchain technology and uses decentralized consensus method to price contributions and provide equity returns for content production areas, enabling content producers, content investors, content filterers and eco-builders to get reasonable incentives and rewards. YOYOW focuses on the underlying technology and large profit distribution settings, and assigns greater autonomy to integrated platforms and developers. Platforms and developers can build their own blockchain-based content products by relying on YOYOW without much blockchain expertise.

Integration of the media platforms can bring the following advantages:

- Realizing the integration of current business + blockchain, introducing a reasonable incentive mechanism, and effectively enhancing user enthusiasm
- Easily realizing the issuance of token assets and getting convenient token monetization channels
- Can directly get YOYOW incentives
- Can share all current users of YOYOW

22.1 Introduction to YOYOW Ecology

22.1.1 User Roles

In the system ecology, in order to balance efficiency and fairness, YOYOW has designed a complete user system, including common users, platforms, committees, and witnesses.

Common Users

Common users are the main participants of the YOYOW network. In the YOYOW network, common users share a unified account system on each platform, and have the rights of transfers, election, etc.

Platforms

The platform is a special kind of user. Common user that deposits a collateral of a certain amount of YOYO tokens can be called a platform. Common users can authorize the platform through the YOYOW wallet, and the authorized platform uses its Tipping authority for cross-site login, content rating, content posting, comments, etc., and common users can also revoke authorization at any time. After integration, the platforms can get token incentives of the YOYOW network.

Committees

The committee is the governing body of the YOYOW network, responsible for initiating and voting on the proposals. The content of the proposals is mainly to adjust dozens of adjustable parameters of the system, such as the rate of various transactions, the highest scoring weight, etc., including as well the block production interval, block rewards, etc. The committee candidates are voted in by the users of YOYOW and the users can withdraw their votes at any time to ensure that the resolutions of the committee truly reflect the wishes of the majority of the token holders.

Witnesses

The witness is the producer of the block, responsible for collecting the various transactions of the broadcast and packing them into the blocks, so the witnesses producing blocks can also get the corresponding YOYO token as returns.

The witnesses include Top witnesses, Standby witnesses, and Miner witnesses.

Under the current mechanism, YOYOW users can fully exercise their rights. Both the committee and the witnesses are elected by votes, and all is decided according to the number of votes. Users can vote for an account that is willing to participate in community building and is willing to contribute to the community. At the same time, each vote is valid for 90 days, which guarantees the vitality of the committee and the witness team.

22.1.2 Private Key and Assets

Private Key System

In the blockchain project, the private key is the top priority of the account. However, when a user does any operation and uses the account private key to sign, it will undoubtedly greatly increase the risk of the account. Considering the different security levels of different authorities, the three-level key system is set in the YOYOW system: **the Owner Key**, **the Active Key**, and **the Secondary Key**. The Owner Key is the key of highest authority, and the Owner Key should not be used unless it is absolutely necessary. The Active Key can operate the balance. The Secondary Key can operate the Tipping, and can also be used for posting, giving likes, login authentication, etc.

Simply speaking:

1. The Owner key is the core. If other low-level keys are lost, you can use the Owner key to reset and update. Therefore, it must be kept properly and once lost, it cannot be retrieved.
2. For fund operations such as transfer, you can use the Active key.
3. The users who want to authorize the platforms only need to grant the Secondary key (Tipping authority), through which the platforms can act as proxy for the users to post, give likes and make small-amount transfers.

In addition to the asset-related 3-level keys, there is also a **Memo Key** used to encrypt and view the note information for the transactions.

Each of the above groups of keys consists of a pair of public and private key pairs. The public key is 53 characters starting with YYW and the private key is 51 characters. Each key can be viewed through [web wallet](#) (see: [yoyow keys tutorial](#)). For example:



Types of Assets

There are three types of assets in the YOYOW network: **YOYO token**, **user-issued assets (UIA)**, and **bonus points**.

YOYO Token

In YOYO's account, the base asset is YOYO token, and there are two places to store: **balanceandtipping**.

- **Balance** has a high level of security, and it is recommended that a large amount of YOYO be stored in the balance.
- **Tipping**, which can be understood as password-free payment, is for small amount of savings and transfer. When the user grants the authority of tipping to the platform, the platform can even have the right to operate "Tipping".

User-Issued AssetsUIA

Users can issue custom assets: users can create Tokens that circulate within their own platforms based on their business. A convenient exchange method between the user-issued assets (UIA) and YOYO will be provided subsequently, sharing convenient monetization channels.

Bonus Points

Bonus Points are rewards for YOYO holders. When your account holds YOYO, Bonus Points will be accumulated over time. There is an upper limit for Bonus Points accumulated, and you have to withdraw Bonus Points manually.

Using Bonus Points

At present, the only use of Bonus Points is for paying fees. You can use Bonus Points to offset fees. 100 Bonus Points can be used as 1 YOYO.

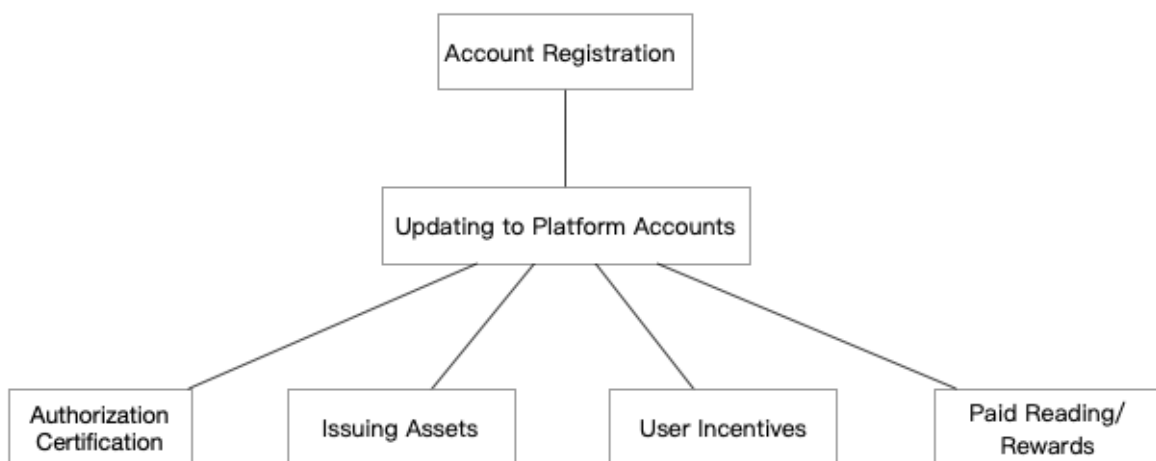
When the “balance” of your account reaches a certain amount of YOYO, the system will automatically accumulate Bonus Points for you. There is a limit to the highest amount of accumulation.

Using Method for Bonus Points

Bonus Points are withdrawn manually. You can manually withdraw available Bonus Points to your account or other accounts of your contact as incentives. Bonus Points can be rented to others for use.

YOYOW introduces Bonus Points, which can reduce the expenditure of common users’ small amount of fees (content release, likes, etc.) and increase the enthusiasm of users. In other blockchain projects, different levels of fees are required for chain operations. For YOYOW, this bonus point system makes it possible for users to operate for free.

22.2 Integration Process



Integration Process

22.2.1 Account Registration

To apply for a YOYOW account, you need to submit a registration page. Access the webpage <https://wallet.yoyow.org/#/create-account>, set the password, and YOYOW will automatically return the account name and account private key. The account name is the number UID (yoyo number), the private key is the Owner key. The private key will not be saved by the system, and will only appear once upon successful registration. It has the highest authority to manage

the account. Please save it properly. It is not recommended to save it to a computer, net-disk or cloud note. You can write it on multiple notes and save it to a few more secure locations, ensuring that they will not be lost or leaked.

Register

Note:

Your password is not stored in the browser or in a central server, please ensure that you remember this password!

[Create Account](#)

[Account Restore](#)

[Account Reset](#)

 **Congratulations, registered successfully!**

YOYOW ID:

228621637

Account Private Key:



Account Private Key owns the supreme authority, can be used for account recovery and password reset!
Private key is NOT stored in the local wallet or browser. Please keep it safe as it cannot be recovered!

I have already backup the Account Private Key

Complete

After completing the registration, you can view your Owner key, Active key, Secondary key and Memo key in Settings–Accounts.

22.2.2 Updating to Platform Accounts

A common account needs to be upgraded to a platform account to obtain the user’s authorization. The upgraded platform currently needs to deposit a collateral of 10,000 YOYO and a fee of 1000 YOYO. This operation is temporarily performed through the YOYOW client. You need to perform the create_platform operation through the interface, and set the name of the platform, the symbol of the Token and the related url address and other extra data, such as the API access interface provided by the platform.

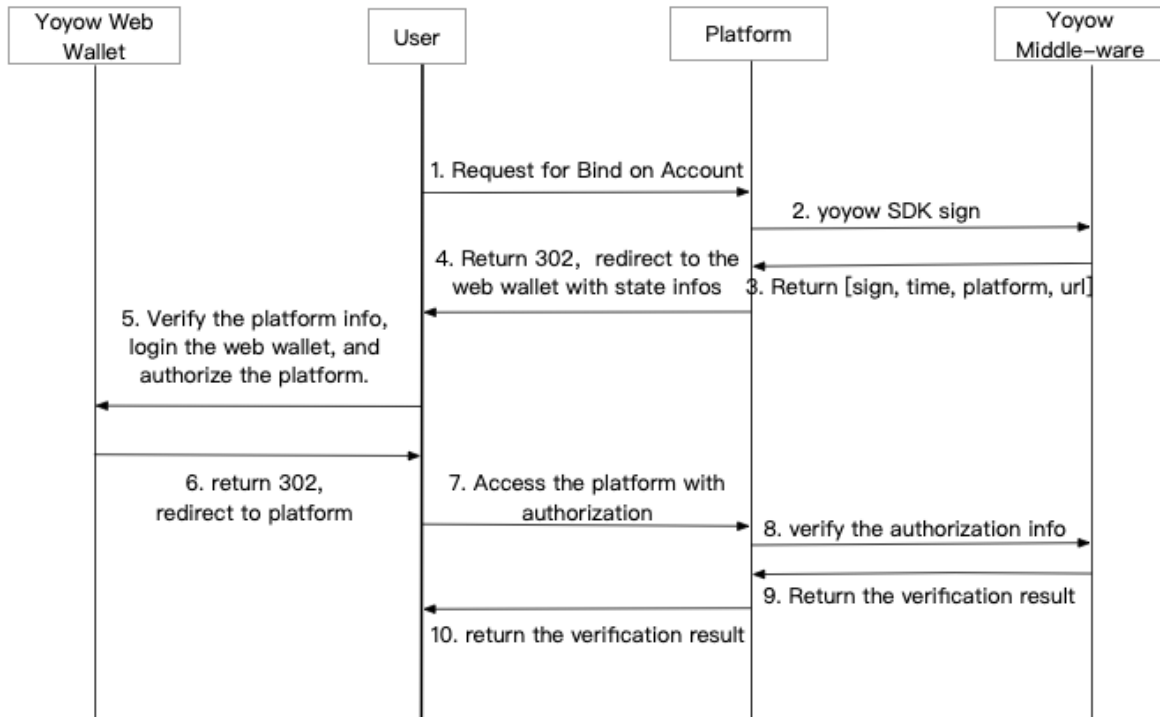
For details on how to deposit the upgraded platform account, see [Creating YOYOW Platform Account from scratch](#).

22.2.3 Authorization Certification

YOYOW provides an authorization certification similar to OAuth. YOYOW provides middleware to facilitate platform integration. The middleware provides interfaces such as signature platform (sign), signature authentication (verify),

and return QR code on signature platform (SignQR) (see: [yoyow-middleware#About Auth](#)). The platform can generate a binding link through the sign or signQR interface in the SDK, and can verify user authorization through the verify interface. Through authorization, the platform can obtain the user's Tipping management authority and login authorization.

Authorization Process



Note:

1. Considering security issues, the YOYOW middleware SDK is strongly recommended to be deployed separately on intranet machines without opening IP and ports.
2. YOYOW middleware SDK requires Tipping authority. When using, remember to modify the default Secondary key and Memo key in the configuration file.

22.2.4 Issuing Assets

The media can issue its own assets as needed, and the number of assets and exchange rates can be set by their own needs. The yoyow platform provides the convenience of circulation and trading for set assets.

The issued assets are temporarily operated by the create_asset function of the wallet client, and the total amount of assets, the transaction rate of the transaction, and the whitelist and blacklist settings need to be specified.

See [YOYOW Wallet API#create_assets](#).

22.2.5 User Incentives

The platform can specify the incentive plan for its own business needs. The target of the incentive can be YOYO or the tokens issued by itself.

Taking Biask as an example, Biask has been integrated to the YOYOW network. In the platform of Biask, the system reward period is one week, and the likes, the like energy and the number of comments obtained by the user's posted

content in a week are regularly counted and a bonus value is calculated, and YOYO token will be issued according to the proportion of the user's bonus value. According to the user's activity and content contribution value, the user will have prestige attributes. If praised by users with higher prestige, the users get higher like energy, and the liked users will also get higher rewards. Users can withdraw YOYO from the web wallet to other places for trading.

22.2.6 Paid Reading/Rewards

Through the YOYOW network, the platform can easily integrate to the system of cryptocurrency rewards/paid reading.

The platform can encourage users to forward transactions between them, and build a platform for paid reading and rewarding content based on YOYO assets or assets issued by itself. YoYow's blockchain service ensures that forwarding transactions between users is secure, non-tamperable, and transparent, thereby motivating users to deliver higher quality content.

More:

1. YOYOW Middleware <https://github.com/yoyow-org/yoyow-middleware>.

Creating YOYOW Platform Account from scratch

This article will take the testnet environment as an example to introduce the operations required to create the platform, such as registering an account in the web wallet and creating a platform through the command line in the cli.

Resources Required:

Official Network

Web Wallet Address <https://wallet.yoyow.org>. Official Network CLI Download <https://github.com/yoyow-org/yoyow-core/releases>

Testnet

Web Wallet Testing Address <http://demo.yoyow.org:8000>. Testnet CLI Download <https://github.com/yoyow-org/yoyow-core-testnet/releases/>.

23.1 1. Creating Accounts, Getting Private Keys

Take the testnet as an example, visit [testnet web wallet address](#) registration or login account:

Register

Note:

Your password is not stored in the browser or in a central server, please ensure that you remember this password!

Create Account

Account Restore

Account Reset

The steps for platform owners to get the private keys of different authority levels are as follows: log in to the wallet -> settings on the left menu -> account -> view authority -> click “display private key” on the right side of the corresponding authority key -> enter the password to display the private key -> copy the private key you see to the configuration.

The screenshot shows a dark-themed settings menu on the left with options: vote, Gateway, Settings (highlighted), Account, Reset, and Backup. The main content area displays the 'Secondary key' section, which includes a 'Click to show QR code' button, a 'Private Key' field with a password mask (5K...), and a 'Show Public Key' button. Below this is the 'Memo key' section, which includes a QR code, a 'Public Key' field with a long alphanumeric string (YYW5VYkqLXqwx...), and a 'Show Private Key' button.

23.2 2. Creating Platforms

It takes at least 11,000 YOYO to create a platform, of which 10000 **is** the minimum collateral deposit **and** 1000 **is** the platform fee (12000 test tokens are given **for** testnet registration)

23.2.1 2.1 Starting the Client Wallet

Take the testnet as an example, download the yoyow-client in the corresponding environment from [Testnet CLI Download](#).

2.1.1 Starting with Parameters

Take Ubuntu as an example, connect the node of the testnet (ws://47.52.155.181:10011)

```
./yoyow_client -s ws://47.52.155.181:10011
```

If it prompts “insufficient authority”

```
sudo chmod a+x *
```

2.1.2 Starting with a Configuration File

Create a wallet.json file under the same path with the cli wallet

write

```
{
  "chain_id": "3505e367fe6cde243f2a1c39bd8e58557e23271dd6cbf4b29a8dc8c44c9af8fe",
  "pending_account_registrations": [],
  "pending_witness_registrations": [],
  "labeled_keys": [],
  "blind_receipts": [],
  "ws_server": "ws://47.52.155.181:10011",
  "ws_user": "",
  "ws_password": ""
}
```

then you can directly run

```
./yoyow_client
```

23.2.2 2.2 Setting Wallet Password

Upon successful connection, it shows

Please use the set_password method to initialize a new wallet before continuing

```
new >>>
```

```
Execute
```

(continues on next page)

(continued from previous page)

```
new >>> set_password your password

Return

set_password your password
null
locked >>>

Execute

locked >>> unlock your password

Return

unlock 123
null
unlocked >>>

means successful unlock
```

23.2.3 2.3 Importing Active Keys

```
unlocked >>> import_key yoyow account uid active keys

For example:

unlocked >>> import_key 120252179 5JwREzpbw62iEcD6J6Wxs2fbn1aSKWQWvGLNCqAEYws31EHD7i4

Return

1937037ms th_a wallet.cpp:820 save_wallet_file ] saving_
↪wallet to file wallet.json
true

If it doesn't return true, please check if your uid and private key are correct.
```

23.2.4 2.4 Creating Platforms

```
Unlocked >>> create_platform yoyow account uid "platform name" collateral amount_
↪currency symbol "platform url address" "platform extra data json string" true

For example:

unlocked >>> create_platform 235145448 "myPlatform" 10000 YOYO "www.example.com" "{}"
↪true

Return

{
  "ref_block_num": 33094,
  "ref_block_prefix": 2124691028,
```

(continues on next page)

(continued from previous page)

```

"expiration": "2018-02-07T08:40:30",
"operations": [[
  20, {
    "fee": {
      "total": {
        "amount": 1029296,
        "asset_id": 0
      },
      "options": {
        "from_csaf": {
          "amount": 1029296,
          "asset_id": 0
        }
      }
    },
    "account": 235145448,
    "pledge": {
      "amount": 1000000000,
      "asset_id": 0
    },
    "name": "myPlatform",
    "url": "www.example.com",
    "extra_data": "{}"
  }
],
"signatures": [
  ↪ "1f08b704dd5ccf7e05e5dec45b06ad41e6382f5dd528e3f644d52ff4fb29c2040507544d5e94b84d77d70edcd68bb35b0
  ↪ "
  ]
}

```

23.2.5 2.5 Updating Platforms

```

Unlocked >>> update_platform yoyow account uid "platform name" collateral amount_
↪ currency symbol "platform url address" "platform extra data json string" true

```

For example:

```

unlocked >>> update_platform 235145448 "newplatformname" 10000 YOYO null null true

```

the way of returning **is** the same **with** creating a platform

The platform name, platform url address, **and** platform extra data are filled **in with** ↪
 ↪ null **if** there **is** no change. As the operation **in** the example, the platform url ↪
 ↪ address **and** extra data will **not** be changed.

23.2.6 2.6 Platform Extra Data Protocol

The extra data of extra_data in the platform attribute is optional content. The ↪
 ↪ relevant information in the YOYOW login protocol specifies the JSON object format ↪
 ↪ string format and follows the format below.

(continues on next page)

(continued from previous page)

```
{
"login":"http://example/login" Platform scan-code login request interface

"description":"platform description" platform description

"image":"http://example.image.jpg" platform imageplatform image displayed in yoyow_
↔app 1.1

"h5url":"http://exampleH5.com" Platform h5 address, skip to the h5 page if the app_
↔hasn't been installed

"packagename":"com.example.app" Platform android jump

"urlscheme":"example://" Platform ios jump

}
```

CHAPTER 24

More Materials

- Web Wallet <https://wallet.yoyow.org/>
- Client Wallet: <https://yoyow.org/client1.html>
- DPoS.Club: Creating the best DPoS technology exchange forum

CHAPTER 25

Indices and tables

- `genindex`
- `modindex`
- `search`