

Tórónet Portal API

v1.1

3rd Edition

Toro Software Development Corp.

Date: May. 13, 2023
Updated June 1, 2023

Contents

1. Overview.....	8
1.1 Blockchain Call Types	8
1.2 Non-Custodial Calls.....	8
1.3 Custodial Calls	8
1.4 API Call Format	10
2. KeyStore Calls	10
2.1 createKey.....	10
2.2 getKey.....	10
2.3 verifyKey.....	11
2.4 updatekeypwd.....	12
2.5 importkey	12
2.6 deletekey	13
3. Blockchain calls	13
3.1 getstatus.....	13
3.2 getblock	13
3.3 gettransaction	14
3.4 getreceipt	14
3.5 getrevertreason.....	14
4. TNS calls.....	15
4.1 getname	15
4.2 getaddr	15
4.3 isnameused	15
4.4 isaddrassigned	15
4.5 isseton	16
4.6 isupdateon.....	16
4.7 isdeleteon.....	16
5. Token calls	17
5.1 getbalance	17
5.2 getname	17
5.3 getsymbol	17
5.4 getdecimal	17
5.5 getminimumallowance.....	18
5.6 getmaximumallowance	18
5.7 getallowance	18
5.8 gettransactionfeefixed	19
5.9 gettransactionfeepercentage	19

5.10	gettransactionfee	19
5.11	getcommissionaddress.....	20
5.12	getcommissionpercentage	20
5.13	getreserve.....	20
5.14	gettoller	20
5.15	gettotalcap.....	21
5.16	gettotalreserving	21
5.17	gettotalcirculating	21
5.18	isenrolled.....	22
5.19	isfrozen	22
5.20	getallowselfenroll	22
5.21	getallowselftransactionfee	23
5.22	getselftransactionfeefixed.....	23
5.23	getselftransactionfeepercentage	23
5.24	getselftransactionfee.....	23
5.25	getallowselfallowance	24
5.26	getselfminimumallowance	24
5.27	getselfmaximumallowance.....	24
5.28	getselfallowance.....	25
5.29	istransferon	25
5.30	isminton.....	25
5.31	isburnon	26
5.32	transfer	26
5.33	calculatetxfee	26
6.	Currency calls.....	26
6.1	List of supported currencies.....	26
6.2	getbalance	27
6.3	getname	27
6.4	getsymbol.....	27
6.5	getdecimal.....	28
6.6	getexchangerate.....	28
6.7	getminimumallowance.....	28
6.8	getmaximumallowance	28
6.9	getallowance	29
6.10	gettranscationfeefixed	29
6.11	gettransactionfeepercentage	29
6.12	gettransactionfee	29

6.13	getcommissionaddress.....	30
6.14	getcommissionpercentage.....	30
6.15	gettorobuyfeefixed.....	30
6.16	gettorobuyfeepercentage.....	31
6.17	gettorobuyfee.....	31
6.18	gettorosellfeefixed.....	31
6.19	gettorosellfeepercentage.....	31
6.20	gettorosellfee.....	32
6.21	getcurrencyimportfeefixed.....	32
6.22	getcurrencyimportfeepercentage.....	32
6.23	getcurrencyimportfee.....	32
6.24	getcurrencyexportfeefixed.....	32
6.25	getcurrencyexportfeepercentage.....	32
6.26	getcurrencyexportfee.....	32
6.27	getreserve.....	32
6.28	gettoller.....	33
6.29	gettotalcap.....	33
6.30	gettotalreserving.....	33
6.31	gettotalcirculating.....	33
6.32	isenrolled.....	33
6.33	isfrozen.....	33
6.34	getallowselfenroll.....	33
6.35	getallowselftransactionfee.....	34
6.36	getselftransactionfeefixed.....	34
6.37	getselftransactionfeepercentage.....	34
6.38	getselftransactionfee.....	34
6.39	getallowselfallowance.....	34
6.40	getselfallowance.....	34
6.41	getallowselftorobuyfee.....	34
6.42	getselftorobuyfeefixed.....	35
6.43	getselftorobuyfeepercentage.....	35
6.44	getselftorobuyfee.....	35
6.45	getallowselftorosellfee.....	35
6.46	getselftorosellfeefixed.....	35
6.47	getselftorosellfeepercentage.....	35
6.48	getselftorosellfee.....	35
6.49	getallowselfcurrencyimportfee.....	36

6.50	getselfcurrencyimportfeefixed	36
6.51	getselfcurrencyimportfeepercentage	36
6.52	getselfcurrencyimportfee	36
6.53	getallowselfcurrencyexportfee	36
6.54	getselfcurrencyexportfeefixed	36
6.55	getselfcurrencyexportfeepercentage	36
6.56	getselfcurrencyexportfee	37
6.57	istransferon	37
6.58	isbuyon	37
6.59	issellon	37
6.60	isimporton	37
6.61	isexporton	37
7	Crypto calls	37
7.1	List of all supported cryptocurrencies	37
7.2	getbalance	38
7.3	getname	38
7.4	getsymbol	38
7.5	getdecimal	38
7.6	getexchangerate	39
7.7	getminimumallowance	39
7.8	getmaximumallowance	39
7.9	getallowance	40
7.10	gettranscationfeefixed	40
7.11	gettransactionfeepercentage	40
7.12	gettransactionfee	40
7.13	getcommissionaddress	41
7.14	getcommissionpercentage	41
7.15	gettorobuyfeefixed	41
7.16	gettorobuyfeepercentage	42
7.17	gettorobuyfee	42
7.18	gettoroselffeefixed	42
7.19	gettorosellfeepercentage	42
7.20	gettorosellfee	43
7.21	getcryptoimportfeefixed	43
7.22	getcryptoimportfeepercentage	43
7.23	getcryptoimportfee	43
7.24	getcryptoexportfeefixed	43

7.25	getcryptoexportfeepercentage	43
7.26	getcryptoexportfee	43
7.27	getreserve.....	43
7.28	gettoller	44
7.29	gettotalcap	44
7.30	gettotalreserving	44
7.31	gettotalcirculating	44
7.32	isenrolled.....	44
7.33	isfrozen	44
7.34	getallowselfenroll	44
7.35	getallowselftransactionfee	45
7.36	getselftransactionfeefixed.....	45
7.37	getselftransactionfeepercentage	45
7.38	getselftransactionfee.....	45
7.39	getallowselfallowance	45
7.40	getselfallowance.....	45
7.41	getallowselftorobuyfee	45
7.42	getselftorobuyfeefixed	46
7.43	getselftorobuyfeepercentage.....	46
7.44	getselftorobuyfee	46
7.45	getallowselftorosellfee	46
7.46	getselftorosellfeefixed.....	46
7.47	getselftorosellfeepercentage	46
7.48	getselftorosellfee.....	46
7.49	getallowselfcryptoimportfee	47
7.50	getselfcryptoimportfeefixed.....	47
7.51	getselfcryptoimportfeepercentage	47
7.52	getselfcryptoimportfee	47
7.53	getallowselfcryptoexportfee	47
7.54	getselfcryptoexportfeefixed	47
7.55	getselfcryptoexportfeepercentage.....	47
7.56	getselfcryptoexportfee.....	48
7.57	istransferon	48
7.58	isbuyon	48
7.59	issellon.....	48
7.60	isimporton	48
7.61	isexporton.....	48

7.62	hasextlink	48
7.63	getnumberofextlink.....	48
7.64	gettextlinkindex	49
7.65	gettextlinkbyindex	49
7.66	iscryptoextlinked	49

1. Overview

In this API, RESTful https calls are provided to execute fundamental Tóronet tasks.

In this manual, both signed (non-custodial) and unsigned (custodial) calls will be introduced:

The current base access point for the RESTful API on the test portal is

<https://testnet.toronet.org/api/>

In this manual, the syntax, arguments, and responses will be introduced.

1.1 Blockchain Call Types

There are two types of blockchain calls:

- 1 Calls that request information from the blockchain. Blockchain information is public and transparent. There are several calls that is provided to access such data. However, while data is publicly available, note that such data could also be encrypted by process or smart contracts; which ensures that certain information can still be made accessible only to owners of such data. Calls that request information do not typically need to be signed to obtain such information.
- 2 Calls that make changes to the blockchain. These calls add data to a block, and needs to be added by the blockchain consensus algorithm. The process pf generating a new block is sometimes called mining. All calls that generate data that needs to be added to the blockchain needs to be signed by an account. The account could be a user account, or a contract account. To sign an account, the account owner needs to have access and ownership to the primary key of the account. Only that primary key can encrypt the information corresponding to the public key. There are two ways this can be accomplished, and they are described below.

1.2 Non-Custodial Calls

For non-custodial calls, the primary keys for the accounts are located on the device of the owner of the account. The account owner accesses that primary key, utilizing software on that device to encrypt the information. The encrypted information is then transmitted to the blockchain network where is will be saved in a pool of potential new transactions or data intended to be included in the next block. The blockchain nodes then pick up that transaction or block of data, verifies that it is duly signed, and if so will include it in a block and process the block as the next block through the consensus procedure of the chain.

Sometimes, the primary key itself is saved in a keyfile, which is then password-protected. The user can then use the password to simply unlock and use it to sign the transaction they intend to send. The primary key itself, in that situation, does not need to be directly accessed or even known by the user – just the password. Note that gaining access to the keyfile, without knowing the password accomplishes absolutely nothing. What makes the process non-custodial is simply the location of the primary key, or it's representative keyfile.

1.3 Custodial Calls

For custodial calls, the keyfile described above is saved on the API server. However, the password to it is still only known to the user. To initiate a transaction, the API software over a secure layer would request the user to use their password to unlock the keyfile and then sign the intended transaction the same way as for custodial transaction. Note that similarly, the keyfile itself is completely useless without the password that can unlock it. This process is similar to the overall blockchain principle which we term data in plain sight secured by sovereign users. Blockchain data are available publicly made possible via encryption, and only those who have ownership or authority over such data can order, decrypt, or modify the data.

1.4 API Call Format

All Toronet API calls use JSON format for the input, as well as the output. Much of the input and output are also formatted as name/value pairs. This becomes clear after reviewing a few examples. In the next few sections, we review all currently available calls including their input, output, and in the case of an error, the various error messages returned.

2. KeyStore Calls

2.1 createKey

This call creates a new blockchain address. The syntax of the call is as follows:

```
{
  "op": "createkey",
  "params":
  [
    {
      "name": "password",
      "value": "Toronet"
    }
  ]
}
```

The argument or attribute of the call are:

Attributes	Type
op	string
name	string constant
value	string

The response for this call is a new public address or account on the blockchain.

Response Attributes	Type
Address	string/address

2.2 getKey

This call retrieves the key of the blockchain address. The payload is as follows:

```
{
  "op": "getKey",
  "params":
  [
    {
      "name": "addr",
      "value": "0x432c7294c152c4f4930373a79e833cd22b9925b3"
    }
  ]
}
```

The arguments of the call are:

Attributes	Type
op	string
name	string constant
value	string

The response for this call is the keystoredata

```
{
  "result": true,
  "keystoredata": {
    "address": "432c7294c152c4f4930373a79e833cd22b9925b3",
    "crypto": {
      "cipher": "aes-128-ctr",
      "ciphertext": "561a6c0998109babb828fa51a54e03557ca6620854ec41b9a99ca3e66d517b21",
      "cipherparams": {
        "iv": "ab4c9793b35edb7315c7cc851a29074e"
      },
      "mac": "17063e95e2b850c4f3714c30e3e5875705e4684b46fdeba0f48aa60167bc57",
      "kdf": "pbkdf2",
      "kdfparams": {
        "c": 262144,
        "dklen": 32,
        "prf": "hmac-sha256",
        "salt": "beee530d83acaa730d4df6857148e83256125f79cf50da37f4a09e38b5ce4fb8"
      }
    },
    "id": "01b9c4ea-8eeb-48f9-8d68-ddc88ed2942d",
    "version": 3
  },
  "message": "keystore record has been found"
}
```

2.3 verifyKey

This call is used to verify the validity of a cryptographic key stored in the keystore. The payload is as follows:

```
{
  "op": "verifykey",
  "params": [
    {
      "name": "addr",
      "value": "0x432c7294c152c4f4930373a79e833cd22b9925b3"
    },
    {
      "name": "pwd",
      "value": "toronet"
    }
  ]
}
```

The response for this call is the result

```
{
  "result": true,
  "message": "keystore record is found and the password has been verified"
}
```

2.4 updatekeypwd

This call is used to update the key stored in the keystore. The payload is as follows:

```
{
  "op": "updatepwd",
  "params": [
    {
      "name": "addr",
      "value": "0x432c7294c152c4f4930373a79e833cd22b9925b3"
    },
    {
      "name": "oldpwd",
      "value": "toronet2"
    },
    {
      "name": "newpwd",
      "value": "toronet"
    }
  ]
}
```

The arguments of the call are:

Attributes	Type
op	string
name	string constant
value	string
oldpwd & pwd	string

2.5 importkey

This is called to import a keystore record by private key and password, The payload is as follows:

```
{ "op": "importkey", "params": [{"name": "prvkey",
"value": "8da4ef21b864d2cc526dbdb2a120bd2874c36c9d0a1fb7f8c63d7f7a8b41de8f"}, {"name": "pwd",
"value": "toronet"}] }
```

The response for this call is:

```
{
  "result": true,
}
```

```
{
  "address": "0x63fac9201494f0bd17b9892b9fae4d52fe3bd377",
  "message": "keystore record has been imported"
}
```

2.6 deletekey

This call is used to delete key from the API server. The payload is as follows:

```
{ "op": "deletekey", "params": [{"name": "addr",
"value": "63fac9201494f0bd17b9892b9fae4d52fe3bd377"}, {"name": "pwd", "value": "toronet"}] }
```

The response for this call is:

```
{
  "result": true,
  "message": "keystore record has been deleted"
}
```

3. Blockchain calls

3.1 getstatus

This call is used to get basic information about the Blockchain.

```
http://testnet.toronet.org/api/blockchain
```

The response for this call is:

```
{
  "result": true,
  "blockchaininfo": {
    "chain": "testnet",
    "chainid": 54321,
    "latestblock": 11575975,
    "datetime": "2023/05/17 10:09:54"
  },
  "message": "current blockchain status has been updated"
}
```

3.2 getblock

This call is used to query the latest block on the Blockchain. The payload is as follows:

```
{ "op": "getblock", "params": [{"name": "id", "value": "latest"}] }
```

The response for this call is:

```
{
  "result": true,

```


4. TNS calls

4.1 getname

This call is used to retrieve the name associated with a specific address; the payload is as follows:

```
{ "op": "getname", "params": [{"name": "addr",  
"value": "0x43F78b342084e370f10e0Cd07d56d95c1728C9D4"}] }
```

The response for this call is:

4.2 getaddr

This call is used to retrieve the address associated with a specific name. The payload is as follows:

```
{ "op": "getaddr", "params": [{"name": "name", "value": "owner"}] }
```

The response for this call is:

```
{  
  "result": true,  
  "address": "0x314EF41554Dc423C88836DCDce55b3f61d1804B1",  
  "message": "tns address for 'owner' is '0x314EF41554Dc423C88836DCDce55b3f61d1804B1'"  
}
```

4.3 isnameused

This call is used to check if a specific name is already used in the TNS, the payload is as follows:

```
{ "op": "isnameused", "params": [{"name": "name", "value": "owner"}] }
```

The response for this call is:

```
{  
  "result": true,  
  "isused": true,  
  "message": "name 'owner' is used in tns"  
}
```

4.4 isaddrassigned

This call is used to check if a specific address has been assigned in the TNS. The payload for this request is as follows:

```
{ "op": "isaddrassigned", "params": [{"name": "addr",  
"value": "0x43F78b342084e370f10e0Cd07d56d95c1728C9D4"}] }
```

The response for this call is:

```
{
  "result": true,
  "isassigned": false,
  "message": "address '0x43F78b342084e370f10e0Cd07d56d95c1728C9D4' is not assigned in tns"
}
```

4.5 isseton

This call is used to check if the TNS is currently enabled

```
{ "op": "isseton", "params": [] }
```

The response for this call is:

```
{
  "result": true,
  "ison": false,
  "message": "client tns name set is off"
}
```

4.6 isupdateon

This call is used to check if the update functionality of the TNS is currently enabled

```
{ "op": "isupdateon", "params": [] }
```

The response of this call is:

```
{
  "result": true,
  "ison": false,
  "message": "client tns name update is off"
}
```

4.7 isdeleteon

This call is used to check if the delete functionality of the TNS is currently enabled

```
{ "op": "isdeleteon", "params": [] }
```

The response for this call is:

```
{
  "result": true,
  "ison": false,
  "message": "client tns name delete is off"
}
```



```
}
```

5. Token calls

5.1 getbalance

This call is used to get The Tòrø balance for an address, the payload is as follows:

```
{ "op": "getbalance", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcdce55b3f61d1804b1"}] }
```

The response for this call is:

```
{  
  "result": true,  
  "balance": "400",  
  "message": "toro balance for '0x314ef41554dc423c88836dcdce55b3f61d1804b1' is '400'"  
}
```

5.2 getname

This call is used to get the token's name.

```
{ "op": "getname", "params": [] }
```

The response for this call is:

```
{  
  "result": true,  
  "name": "Toro",  
  "message": "token name is 'Toro'"  
}
```

5.3 getsymbol

This call is used to get the token's symbol

```
{ "op": "getsymbol", "params": [] }
```

The response for this call is:

```
{  
  "result": true,  
  "symbol": "TORO",  
  "message": "token symbol is 'TORO'"  
}
```

5.4 getdecimal

This call is used to get the token's decimal

```
{ "op": "getdecimal", "params": [] }
```

The response for this call is:

```
{  
  "result": true,  
  "decimal": "18",  
  "message": "token decimal is '18'"  
}
```

5.5 getminimumallowance

This call is used to get the token's minimum transfer allowance

```
{ "op": "getminimumallowance", "params": [] }
```

The response for this call is:

```
{  
  "result": true,  
  "minimumallowance": "0",  
  "message": "toro minimum transfer allowance is '0'"  
}
```

5.6 getmaximumallowance

This call is used to get the token's maximum transfer allowance

```
{ "op": "getmaximumallowance", "params": [] }
```

The response for this call is:

```
{  
  "result": true,  
  "maximumallowance": "10000",  
  "message": "toro maximum transfer allowance is '10000'"  
}
```

5.7 getallowance

This call is used to get the maximum / minimum transfer allowance

```
{ "op": "getallowance", "params": [] }
```

The response for this call is:

```
{
  "result": true,
  "minimumallowance": "0",
  "maximumallowance": "10000",
  "message": "toro maximum / minimum transfer allowance are '10000 / 0'"
}
```

5.8 gettransactionfeefixed

This call is used to get the transaction (this is fixed)

```
{ "op": "gettransactionfeefixed", "params": [] }
```

The response for this call is:

```
{
  "result": true,
  "txfeefixed": "0",
  "message": "toro fixed transaction fee is '0'"
}
```

5.9 gettransactionfeepercentage

This call is used to get the transaction fee in percentage (%)

```
{ "op": "gettransactionfeepercentage", "params": [] }
```

The sample response for this response is:

```
{
  "result": true,
  "txfeepercentage": "0",
  "message": "toro percentage transaction fee is '0%'"
}
```

5.10 gettransactionfee

This call is used to get the transaction fee

```
{ "op": "gettransactionfee", "params": [] }
```

The sample response for this call is:

```
{
  "result": true,
  "txfeefixed": "0",
  "txfeepercentage": "0",
  "message": "toro fixed / percentage transfer fee are '0' / '0%'"
}
```

```
}
```

5.11 getcommissionaddress

This call is used to get the commission address

```
{ "op": "getcommissionaddress", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "commissionaddress": "0xa231BB16803d8F7dcb6885B04183c9E71F4cdDF3",  
  "message": "commission address is '0xa231BB16803d8F7dcb6885B04183c9E71F4cdDF3'"  
}
```

5.12 getcommissionpercentage

This call is used to get the commission percentage (%)

```
{ "op": "getcommissionpercentage", "params": [] }
```

The response for this call is:

```
{  
  "result": true,  
  "commissionpercentage": "0",  
  "message": "commission percentage is '0 %'"  
}
```

5.13 getreserve

This call is used to get the reserve address

```
{ "op": "getreserve", "params": [] }
```

The response for this call is:

```
{  
  "result": true,  
  "reserve": "0xf3CDfC4a1DCE2D98FF878971626B798279954c43",  
  "message": "reserve address is '0xf3CDfC4a1DCE2D98FF878971626B798279954c43'"  
}
```

5.14 gettoller

This call is used to get the toller address

```
"op": "gettoller", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "toller": "0xa231BB16803d8F7dcb6885B04183c9E71F4cdDF3",  
  "message": "toller address is '0xa231BB16803d8F7dcb6885B04183c9E71F4cdDF3'"  
}
```

5.15 gettotalcap

This call is used to get the token's total cap

```
{ "op": "gettotalcap", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "totalcap": "0",  
  "message": "toro total cap is '0'"  
}
```

5.16 gettotalreserving

This call is used to get the total token that is currently in reserve

```
{ "op": "gettotalreserving", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "totalreserving": "71987.274205605201720169",  
  "message": "toro total reserving is '71987.274205605201720169'"  
}
```

5.17 gettotalcirculating

This call is used to get to total token that is currently circulating

```
{ "op": "gettotalcirculating", "params": [] }
```

The sample response for this call is:

```
{
```

```
{
  "result": true,
  "totalcirculating": "38412.725794394798279831",
  "message": "toro total circulating is '38412.725794394798279831'"
}
```

5.18 isenrolled

This call is used to check if a wallet address has been enrolled for Toro, the payload is as follows:

```
{ "op": "isenrolled", "params": [{"name": "addr",
"value": "0x314ef41554dc423c88836dcce55b3f61d1804b1"}] }
```

The sample response is:

```
{
  "result": true,
  "isenrolled": true,
  "message": "'0x314ef41554dc423c88836dcce55b3f61d1804b1' is enrolled for toro"
}
```

5.19 isfrozen

This call is used to check if a wallet address is frozen, the payload is as follows:

```
{ "op": "isfrozen", "params": [{"name": "addr",
"value": "0x314ef41554dc423c88836dcce55b3f61d1804b1"}] }
```

The sample response for this call is:

```
{
  "result": true,
  "isfrozen": false,
  "message": "'0x314ef41554dc423c88836dcce55b3f61d1804b1' is not frozen"
}
```

5.20 getallowselfenroll

This call is used to check if automatic Toro enrollment is allowed

```
{ "op": "getallowselfenroll", "params": [] }
```

The sample response for this call is:

```
{
  "result": true,
  "allowselfenroll": true,
  "message": "automatic toro enrollment is allowed"
}
```

5.21 getallowselftransactionfee

This call is used to check if account specified transaction fee is allowed, the payload is as follows:

```
{ "op": "getallowselftransactionfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcce55b3f61d1804b1"}] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "allowselftransactionfee": false,  
  "message": "account specified transaction fee is not set for  
'0x314ef41554dc423c88836dcce55b3f61d1804b1'"  
}
```

5.22 getselftransactionfeefixed

This call is used to check if account specified transaction fee is allowed, the payload is as follows:

```
{ "op": "getselftransactionfeefixed", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcce55b3f61d1804b1"}] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "selftxfeefixed": "0",  
  "message": "toro account specified fixed transaction fee for  
'0x314ef41554dc423c88836dcce55b3f61d1804b1' is '0'"  
}
```

5.23 getselftransactionfeepercentage

This call gets the account's specified transaction fee percentage, the payload is as follows:

```
{ "op": "getselftransactionfeepercentage", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcce55b3f61d1804b1"}] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "selftxfeepercentage": "0",  
  "message": "toro account specified percentage transaction fee for  
'0x314ef41554dc423c88836dcce55b3f61d1804b1' is '0 %'"  
}
```

5.24 getselftransactionfee

This call is used to get an account's specified fixed/percentage transaction fee, the payload is as follows:

```
{ "op": "getselftransactionfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcdce55b3f61d1804b1"}] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "selftxfeefixed": "0",  
  "selftxfeepercentage": "0",  
  "message": "toro account specified fixed / percentage transaction fee for  
'0x314ef41554dc423c88836dcdce55b3f61d1804b1' are '0' / '0%'"  
}
```

5.25 getallowselfallowance

This call is used to check if specified transfer allowance is set for a wallet address, the payload is as follows:

```
{ "op": "getallowselfallowance", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcdce55b3f61d1804b1"}] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "allowselfallowance": false,  
  "message": "account specified transfer allowance is not set for  
'0x314ef41554dc423c88836dcdce55b3f61d1804b1'"  
}
```

5.26 getselfminimumallowance

This call is used to get the specified minimum transfer allowance for an account, the payload is as follows:

```
{ "op": "getselfminimumallowance", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcdce55b3f61d1804b1"}] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "selfminimumallowance": "0",  
  "message": "toro account specified minimum transfer allowance for  
'0x314ef41554dc423c88836dcdce55b3f61d1804b1' is '0'"  
}
```

5.27 getselfmaximumallowance

This call is used to get the specified maximum transfer allowance for an account, the payload is as follows:

```
{ "op": "getselfmaximumallowance", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcdce55b3f61d1804b1"}] }
```


The sample response for this call is:

```
{
  "result": true,
  "selfmaximumallowance": "0",
  "message": "toro account specified maximum transfer allowance for
'0x314ef41554dc423c88836dcce55b3f61d1804b1' is '0'"
}
```

5.28 getselfallowance

This call is used to get the specified minimum/maximum transfer allowance for an account, the payload is as follows:

```
{ "op": "getselfallowance", "params": [{"name": "addr",
"value": "0x314ef41554dc423c88836dcce55b3f61d1804b1"}] }
```

The sample response for this call is:

```
{
  "result": true,
  "selfminimumallowance": "0",
  "selfmaximumallowance": "0",
  "message": "toro account specified minimum / maximum transfer allowance for
'0x314ef41554dc423c88836dcce55b3f61d1804b1' are '0 / 0'"
}
```

5.29 istransferon

This call is used to check if Toro transfer is allowed

```
{ "op": "istransferon", "params": [] }
```

The sample response for this call is:

```
{
  "result": true,
  "istransferon": true,
  "message": "toro transfer is allowed"
}
```

5.30 isminton

This call is used to check if Toro min is allowed

```
{ "op": "isminton", "params": [] }
```

The sample response for this call is:

```
{
  "result": true,
```

```
"isminton": true,  
"message": "toro mint is allowed"  
}
```

5.31 isburnnon

This call is used to check if Toro burn is allowed

```
{ "op": "isburnnon", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "isburnnon": true,  
  "message": "toro burn is allowed"  
}
```

5.32 transfer

This call is used to transfer Toro token from one address to the other

```
{ "op": "transfer", "params": [{"name": "client",  
"value": "0xf3cdfc4a1dce2d98ff878971626b798279954c43"}, {"name": "clientpwd",  
"value": "toronet"}, {"name": "to", "value": "0x314ef41554dc423c88836dcdce55b3f61d1804b1"},  
{"name": "val", "value": "2"}] }
```

5.33 calculatetxfee

This call is used to calculate transaction fee

```
{ "op": "calculatetxfee", "params": [{"name": "client",  
"value": "0xf3cdfc4a1dce2d98ff878971626b798279954c43"}, {"name": "val", "value": "2"}] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "fee": "0",  
  "message": "transaction fee is '0' toro"  
}
```

6. Currency calls

6.1 List of supported currencies

The currencies that are currently supported on the Toronet Blockchain are:

- dollar
- naira
- euro

- pound
- egb
- ksh
- zar

all API examples are for Dollar. To obtain the corresponding API calls for naira, euro and other currencies, simply change the "/dollar/" to "/naira/" or "/euro/" or the corresponding currency in given API examples

```
http://testnet.toronet.org/api/currency/dollar/
```

6.2 getbalance

This call is used to get the balance of an address, the payload for this call is as follow:

```
{ "op": "getbalance", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcdce55b3f61d1804b1"}] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "balance": "0",  
  "message": "dollar balance for '0x314ef41554dc423c88836dcdce55b3f61d1804b1' is '0'"  
}
```

6.3 getname

This call is used to get name of the currency

```
{ "op": "getname", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "name": "Dollar",  
  "message": "currency name is 'Dollar'"  
}
```

6.4 getsymbol

This call is used to get the symbol of the currency

```
{ "op": "getsymbol", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "symbol": "USD",  
  "message": "currency symbol is 'USD'"  
}
```

6.5 getdecimal

This call is used to get the currency decimal

```
{ "op": "getdecimal", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "decimal": "18",  
  "message": "currency decimal is '18'"  
}
```

6.6 getexchangerate

This call is used to get the current exchange rate between Toro token and a currency

```
{ "op": "getexchangerate", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "exchangerate": "1",  
  "message": "toro / dollar exchange rate is '1'"  
}
```

6.7 getminimumallowance

This call is used to get the minimum transfer allowance for the currency

```
{ "op": "getminimumallowance", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "minimumallowance": "0",  
  "message": "dollar minimum transfer allowance is '0'"  
}
```

6.8 getmaximumallowance

This call is used to get the maximum transfer allowance for the currency

```
{ "op": "getmaximumallowance", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
}
```

```
    "maximumallowance":  
'115792089237316195423570985008687907853269984665640564039457.584007913129639935',  
    "message": "dollar maximum transfer allowance is  
'115792089237316195423570985008687907853269984665640564039457.584007913129639935'"  
}
```

6.9 getallowance

This call is used to get the maximum and minimum transfer allowance for a currency

```
{ "op": "getallowance", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "minimumallowance": "0",  
  "maximumallowance":  
'115792089237316195423570985008687907853269984665640564039457.584007913129639935',  
  "message": "dollar maximum / minimum transfer allowance are  
'115792089237316195423570985008687907853269984665640564039457.584007913129639935 / 0'"  
}
```

6.10 gettransactionfeefixed

This call is used to get the fixed transaction fee

```
{ "op": "gettransactionfeefixed", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "txfeefixed": "0",  
  "message": "dollar fixed transaction fee is '0'"  
}
```

6.11 gettransactionfeepercentage

This call is used to get the transaction fee percentage

```
{ "op": "gettransactionfeepercentage", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "txfeepercentage": "0",  
  "message": "dollar percentage transaction fee is '0%'"  
}
```

6.12 gettransactionfee

This call is used to get the transaction fixed fee and transaction fee percentage

```
{ "op": "gettransactionfee", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "txfeefixed": "0",  
  "txfeepercentage": "0",  
  "message": "dollar fixed / percentage transfer fee are '0' / '0%'"  
}
```

6.13 getcommissionaddress

This call is used to get the commission address

```
{ "op": "getcommissionaddress", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "commissionaddress": "0xcd6B225362bc99cDf7c4FEeA23f13602A281445F",  
  "message": "commission address is '0xcd6B225362bc99cDf7c4FEeA23f13602A281445F'"  
}
```

6.14 getcommissionpercentage

This call is used to get the commission percentage

```
{ "op": "getcommissionpercentage", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "commissionpercentage": "0",  
  "message": "commission percentage is '0 %'"  
}
```

6.15 gettorobuyfeefixed

This call is used to get the currency fixed toro buy fee

```
{ "op": "gettorobuyfeefixed", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "txfeefixed": "0",  
  "message": "dollar fixed toro buy fee is '0'"  
}
```

```
}
```

6.16 gettorobuyfeepercentage

This call is used to get the currency percentage toro buy fee

```
{ "op": "gettorobuyfeepercentage", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "txfeepercentage": "0",  
  "message": "dollar percentage toro buy fee is '0%'"  
}
```

6.17 gettorobuyfee

This call is used to get the fixed and percentage toro buy fee

```
{ "op": "gettorobuyfee", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "txfeefixed": "0",  
  "txfeepercentage": "0",  
  "message": "dollar fixed / percentage toro buy fee are '0' / '0%'"  
}
```

6.18 gettorosellfeefixed

This call is used to get the currency fixed toro sell fee

```
{ "op": "gettorosellfeefixed", "params": [] }
```

This sample response for this call is:

```
{  
  "result": true,  
  "txfeefixed": "0",  
  "message": "dollar fixed toro sell fee is '0'"  
}
```

6.19 gettorosellfeepercentage

This call is used to get the currency toro sell percentage

```
{ "op": "gettorosellfeepercentage", "params": [] }
```

6.20 gettorosellfee

This call is used to get the toro sell fee and toro sell fee percentage

```
{ "op": "gettorosellfee", "params": [] }
```

6.21 getcurrencyimportfeefixed

This call is used to get the currency import fee (fixed)

```
{ "op": "getcurrencyimportfeefixed", "params": [] }
```

6.22 getcurrencyimportfeepercentage

This call is used to get the currency import fee in percentage

```
{ "op": "getcurrencyimportfeepercentage", "params": [] }
```

6.23 getcurrencyimportfee

This call is used to get the fixed / percentage currency import fee

```
{ "op": "getcurrencyimportfee", "params": [] }
```

6.24 getcurrencyexportfeefixed

This call is used to get the currency export fee (fixed)

```
{ "op": "getcurrencyexportfeefixed", "params": [] }
```

6.25 getcurrencyexportfeepercentage

This call is used to get the currency export fee in percentage

```
{ "op": "getcurrencyexportfeepercentage", "params": [] }
```

6.26 getcurrencyexportfee

This call is used to get the fixed/percentage currency export fee

```
{ "op": "getcurrencyexportfee", "params": [] }
```

6.27 getreserve

This call is used to get the currency in reserve


```
{ "op": "getreserve", "params": [] }
```

6.28 gettoller

This call is used to get the toller address

```
{ "op": "gettoller", "params": [] }
```

6.29 gettotalcap

This call is used to get the currency's total cap

```
{ "op": "gettotalcap", "params": [] }
```

6.30 gettotalreserving

This call is used to get the total currency that is currently in reserve

```
{ "op": "gettotalreserving", "params": [] }
```

6.31 gettotalcirculating

This call is used to get the total currency that is currently circulating

```
{ "op": "gettotalcirculating", "params": [] }
```

6.32 isenrolled

This call is used to check if an address is currently enrolled for this currency, the payload is as follows:

```
{ "op": "isenrolled", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.33 isfrozen

This call is used to check if the currency of an address is currently frozen, the payload is as follows:

```
{ "op": "isfrozen", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.34 getallowselfenroll

This call is used to check if automatic currency enrollment is allowed

```
{ "op": "getallowselfenroll", "params": [] }
```

6.35 getallowselftransactionfee

This call is used to check if account specified transaction fee is allowed, the payload is as follows:

```
{ "op": "getallowselftransactionfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.36 getselftransactionfeefixed

This call is used to get the transaction fee (fixed), the payload is follows:

```
{ "op": "getselftransactionfeefixed", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.37 getselftransactionfeepercentage

This call is used to get the transaction fee in percentage, the payload is as follows:

```
{ "op": "getselftransactionfeepercentage", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.38 getselftransactionfee

This call is used to get the fixed and percentage transaction fee, the payload is as follows:

```
{ "op": "getselftransactionfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.39 getallowselfallowance

This call is used to check if specified transfer allowance is set for a wallet address, the payload is as follows:

```
{ "op": "getallowselfallowance", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.40 getselfallowance

This call is used to get the minimum and maximum transfer allowance set for an address in the blockchain, the payload is as follows:

```
{ "op": "getselfallowance", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.41 getallowselftorobuyfee

This call is used to check if account specified Toro buy fee has been set, the payload is as follows:

```
{ "op": "getallowselftorobuyfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.42 getselftorobuyfeefixed

This call is used to check the specified dollar fixed Toro buy fee for an address, the payload is as follows:

```
{ "op": "getselftorobuyfeefixed", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.43 getselftorobuyfeepercentage

This call is used to check the specified percentage for dollar/Toro buy fee for an address, the payload is as follows:

```
{ "op": "getselftorobuyfeepercentage", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.44 getselftorobuyfee

This call is used to check the account specified Toro buy fee, the payload is as follows:

```
{ "op": "getselftorobuyfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.45 getallowselftorosellfee

This call is used to check if account specified Toro sell fee has been set, the payload is as follows:

```
{ "op": "getallowselftorosellfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.46 getselftorosellfeefixed

This call is used to check the specified dollar fixed Toro sell fee for an address, the payload is as follows:

```
{ "op": "getselftorosellfeefixed", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.47 getselftorosellfeepercentage

This call is used to check the specified percentage for dollar/Toro sell fee for an address, the payload is as follows:

```
{ "op": "getselftorosellfeepercentage", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.48 getselftorosellfee

This call is used to check the account specified Toro sell fee; the payload is as follows:

```
{ "op": "getselftorosellfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.49 getallowselfcurrencyimportfee

This call is used to check if the account specified import fee has been set for an address, the payload is as follows:

```
{ "op": "getallowselfcurrencyimportfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.50 getselfcurrencyimportfeefixed

This call is used to check the fixed currency import fee for an address, the payload is as follows:

```
{ "op": "getselfcurrencyimportfeefixed", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.51 getselfcurrencyimportfeepercentage

This call is used to check the currency import fee in percentage, the payload is as follows:

```
{ "op": "getselfcurrencyimportfeepercentage", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.52 getselfcurrencyimportfee

This call is used to get the specified fixed and percentage import fee for an address, the payload is as follows:

```
{ "op": "getselfcurrencyimportfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.53 getallowselfcurrencyexportfee

This call is used to check if a specified fixed and percentage export fee has been set for an address, the payload is as follows:

```
{ "op": "getallowselfcurrencyexportfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.54 getselfcurrencyexportfeefixed

This call is used to get the currency fixed export fee for an address, the payload is as follows:

```
{ "op": "getselfcurrencyexportfeefixed", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.55 getselfcurrencyexportfeepercentage

This call is used to get the currency export fee percentage for an address, the payload is as follows:

```
{ "op": "getselfcurrencyexportfeepercentage", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

6.56 getselfcurrencyexportfee

This call is used to get the account specified fixed and percentage export fee for an address, the payload is as follows:

```
{ "op": "getselfcurrencyexportfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcdce55b3f61d1804b1"}] }
```

6.57 istransferon

This call is used to check if currency transfer is allowed

```
{ "op": "istransferon", "params": [] }
```

6.58 isbuyon

This call is used to check if buying of currency is allowed

```
{ "op": "isbuyon", "params": [] }
```

6.59 issellon

This call is used to check if selling of currency is allowed

```
{ "op": "issellon", "params": [] }
```

6.60 isimporton

This call is used to check if currency import is allowed

```
{ "op": "isimporton", "params": [] }
```

6.61 isexporton

This call is used to check if currency export is on

```
{ "op": "isexporton", "params": [] }
```

7 Crypto calls

7.1 List of all supported cryptocurrencies

The currencies that are currently supported on the Toronet Blockchain are:

- Ethereum (eth)

All API examples are currently for Ethereum.

```
http://testnet.toronet.org/api/crypto/eth/
```

7.2 getbalance

This call is used to get the balance of an address, the payload for this call is as follow:

```
{ "op": "getbalance", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "balance": "0",  
  "message": "eth balance for '0x314ef41554dc423c88836dcfce55b3f61d1804b1' is '0'"  
}
```

7.3 getname

This call is used to get name of the currency

```
{ "op": "getname", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "name": "Ethereum",  
  "message": "currency name is 'Ethereum'"  
}
```

7.4 getsymbol

This call is used to get the symbol of the currency

```
{ "op": "getsymbol", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "symbol": "eth",  
  "message": "currency symbol is 'eth'"  
}
```

7.5 getdecimal

This call is used to get the currency decimal

```
{ "op": "getdecimal", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "decimal": "18",  
  "message": "currency decimal is '18'"  
}
```

7.6 getexchangerate

This call is used to get the current exchange rate between Toro token and a currency

```
{ "op": "getexchangerate", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "exchangerate": "1",  
  "message": "toro / eth exchange rate is '1'"  
}
```

7.7 getminimumallowance

This call is used to get the minimum transfer allowance for the currency

```
{ "op": "getminimumallowance", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "minimumallowance": "0",  
  "message": "eth minimum transfer allowance is '0'"  
}
```

7.8 getmaximumallowance

This call is used to get the maximum transfer allowance for the crypto currency

```
{ "op": "getmaximumallowance", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "maximumallowance":  
  "115792089237316195423570985008687907853269984665640564039457.584007913129639935",  
  "message": "eth maximum transfer allowance is  
'115792089237316195423570985008687907853269984665640564039457.584007913129639935'"  
}
```

```
}
```

7.9 getallowance

This call is used to get the maximum and minimum transfer allowance for a crypto

```
{ "op": "getallowance", "params": [] }
```

The sample response for this call is:

```
{
  "result": true,
  "minimumallowance": "0",
  "maximumallowance":
  "115792089237316195423570985008687907853269984665640564039457.584007913129639935",
  "message": "eth maximum / minimum transfer allowance are
  '115792089237316195423570985008687907853269984665640564039457.584007913129639935 / 0'"
}
```

7.10 gettransactionfeefixed

This call is used to get the fixed transaction fee

```
{ "op": "gettransactionfeefixed", "params": [] }
```

The sample response for this call is:

```
{
  "result": true,
  "txfeefixed": "0",
  "message": "eth fixed transaction fee is '0'"
}
```

7.11 gettransactionfeepercentage

This call is used to get the transaction fee percentage

```
{ "op": "gettransactionfeepercentage", "params": [] }
```

The sample response for this call is:

```
{
  "result": true,
  "txfeepercentage": "0",
  "message": "eth percentage transaction fee is '0%'"
}
```

7.12 gettransactionfee

This call is used to get the transaction fixed fee and transaction fee percentage

```
{ "op": "gettransactionfee", "params": [] }
```


The sample response for this call is:

```
{
  "result": true,
  "txfeefixed": "0",
  "txfeepercentage": "0",
  "message": "eth fixed / percentage transfer fee are '0' / '0%'"
}
```

7.13 getcommissionaddress

This call is used to get the commission address

```
{ "op": "getcommissionaddress", "params": [] }
```

The sample response for this call is:

```
{
  "result": true,
  "commissionaddress": "0xcd6B225362bc99cDf7c4FEeA23f13602A281445F",
  "message": "commission address is '0xcd6B225362bc99cDf7c4FEeA23f13602A281445F'"
}
```

7.14 getcommissionpercentage

This call is used to get the commission percentage

```
{ "op": "getcommissionpercentage", "params": [] }
```

The sample response for this call is:

```
{
  "result": true,
  "commissionpercentage": "0",
  "message": "commission percentage is '0 %'"
}
```

7.15 gettorobuyfeefixed

This call is used to get the crypto fixed toro buy fee

```
{ "op": "gettorobuyfeefixed", "params": [] }
```

The sample response for this call is:

```
{
  "result": true,
  "txfeefixed": "0",
  "message": "eth fixed toro buy fee is '0'"
}
```

7.16 gettorobuyfeepercentage

This call is used to get the currency percentage toro buy fee

```
{ "op": "gettorobuyfeepercentage", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "txfeepercentage": "0",  
  "message": "eth percentage toro buy fee is '0%'"  
}
```

7.17 gettorobuyfee

This call is used to get the fixed and percentage toro buy fee

```
{ "op": "gettorobuyfee", "params": [] }
```

The sample response for this call is:

```
{  
  "result": true,  
  "txfeefixed": "0",  
  "txfeepercentage": "0",  
  "message": "eth fixed / percentage toro buy fee are '0' / '0%'"  
}
```

7.18 gettorosellfeefixed

This call is used to get the currency fixed toro sell fee

```
{ "op": "gettorosellfeefixed", "params": [] }
```

This sample response for this call is:

```
{  
  "result": true,  
  "txfeefixed": "0",  
  "message": "eth fixed toro sell fee is '0'"  
}
```

7.19 gettorosellfeepercentage

This call is used to get the crypto currency toro sell percentage

```
{ "op": "gettorosellfeepercentage", "params": [] }
```

7.20 gettorosellfee

This call is used to get the toro sell fee and toro sell fee percentage

```
{ "op": "gettorosellfee", "params": [] }
```

7.21 getcryptoimportfeefixed

This call is used to get the crypto currency import fee (fixed)

```
{ "op": "getcryptoimportfeefixed", "params": [] }
```

7.22 getcryptoimportfeepercentage

This call is used to get the crypto currency import fee in percentage

```
{ "op": "getcryptoimportfeepercentage", "params": [] }
```

7.23 getcryptoimportfee

This call is used to get the fixed / percentage crypto currency import fee

```
{ "op": "getcryptoimportfee", "params": [] }
```

7.24 getcryptoexportfeefixed

This call is used to get the currency export fee (fixed)

```
{ "op": "getcryptoexportfeefixed", "params": [] }
```

7.25 getcryptoexportfeepercentage

This call is used to get the currency export fee in percentage

```
{ "op": "getcryptoexportfeepercentage", "params": [] }
```

7.26 getcryptoexportfee

This call is used to get the fixed/percentage currency export fee

```
{ "op": "getcryptoexportfee", "params": [] }
```

7.27 getreserve

This call is used to get the crypto currency in reserve

```
{ "op": "getreserve", "params": [] }
```

7.28 gettoller

This call is used to get the toller address

```
{ "op": "gettoller", "params": [] }
```

7.29 gettotalcap

This call is used to get the cryptocurrency's total cap

```
{ "op": "gettotalcap", "params": [] }
```

7.30 gettotalreserving

This call is used to get the total cryptocurrency that is currently in reserve

```
{ "op": "gettotalreserving", "params": [] }
```

7.31 gettotalcirculating

This call is used to get the total currency that is currently circulating

```
{ "op": "gettotalcirculating", "params": [] }
```

7.32 isenrolled

This call is used to check if an address is currently enrolled for this currency, the payload is as follows:

```
{ "op": "isenrolled", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcce55b3f61d1804b1"}] }
```

7.33 isfrozen

This call is used to check if the currency of an address is currently frozen, the payload is as follows:

```
{ "op": "isfrozen", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcce55b3f61d1804b1"}] }
```

7.34 getallowselfenroll

This call is used to check if automatic currency enrollment is allowed

```
{ "op": "getallowselfenroll", "params": [] }
```

7.35 getallowselftransactionfee

This call is used to check if account specified transaction fee is allowed, the payload is as follows:

```
{ "op": "getallowselftransactionfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

7.36 getselftransactionfeefixed

This call is used to get the transaction fee (fixed), the payload is follows:

```
{ "op": "getselftransactionfeefixed", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

7.37 getselftransactionfeepercentage

This call is used to get the transaction fee in percentage, the payload is as follows:

```
{ "op": "getselftransactionfeepercentage", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

7.38 getselftransactionfee

This call is used to get the fixed and percentage transaction fee, the payload is as follows:

```
{ "op": "getselftransactionfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

7.39 getallowselfallowance

This call is used to check if specified transfer allowance is set for a wallet address, the payload is as follows:

```
{ "op": "getallowselfallowance", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

7.40 getselfallowance

This call is used to get the minimum and maximum transfer allowance set for an address in the blockchain, the payload is as follows:

```
{ "op": "getselfallowance", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

7.41 getallowselftorobuyfee

This call is used to check if account specified Toro buy fee has been set, the payload is as follows:

```
{ "op": "getallowselftorobuyfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

7.42 getselftorobuyfeefixed

This call is used to check the specified dollar fixed Toro buy fee for an address, the payload is as follows:

```
{ "op": "getselftorobuyfeefixed", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

7.43 getselftorobuyfeepercentage

This call is used to check the specified percentage for dollar/Toro buy fee for an address, the payload is as follows:

```
{ "op": "getselftorobuyfeepercentage", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

7.44 getselftorobuyfee

This call is used to check the account specified Toro buy fee, the payload is as follows:

```
{ "op": "getselftorobuyfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

7.45 getallowselftorosellfee

This call is used to check if account specified Toro sell fee has been set, the payload is as follows:

```
{ "op": "getallowselftorosellfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

7.46 getselftorosellfeefixed

This call is used to check the specified dollar fixed Toro sell fee for an address, the payload is as follows:

```
{ "op": "getselftorosellfeefixed", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

7.47 getselftorosellfeepercentage

This call is used to check the specified percentage for dollar/Toro sell fee for an address, the payload is as follows:

```
{ "op": "getselftorosellfeepercentage", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

7.48 getselftorosellfee

This call is used to check the account specified Toro sell fee; the payload is as follows:

```
{ "op": "getselftorosellfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

7.49 getallowselfcryptoimportfee

This call is used to check if the account specified import fee has been set for an address, the payload is as follows:

```
{ "op": "getallowselfcryptoimportfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

7.50 getselfcryptoimportfeefixed

This call is used to check the fixed currency import fee for an address, the payload is as follows:

```
{ "op": "getselfcryptoimportfeefixed", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

7.51 getselfcryptoimportfeepercentage

This call is used to check the currency import fee in percentage, the payload is as follows:

```
{ "op": "getselfcryptoimportfeepercentage", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

7.52 getselfcryptoimportfee

This call is used to get the specified fixed and percentage import fee for an address, the payload is as follows:

```
{ "op": "getselfcryptoimportfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

7.53 getallowselfcryptoexportfee

This call is used to check if a specified fixed and percentage export fee has been set for an address, the payload is as follows:

```
{ "op": "getallowselfcryptoexportfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

7.54 getselfcryptoexportfeefixed

This call is used to get the currency fixed export fee for an address, the payload is as follows:

```
{ "op": "getselfcryptoexportfeefixed", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

7.55 getselfcryptoexportfeepercentage

This call is used to get the currency export fee percentage for an address, the payload is as follows:

```
{ "op": "getselfcryptoexportfeepercentage", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcfce55b3f61d1804b1"}] }
```

7.56 getselfcryptoexportfee

This call is used to get the account specified fixed and percentage export fee for an address, the payload is as follows:

```
{ "op": "getselfcryptoexportfee", "params": [{"name": "addr",  
"value": "0x314ef41554dc423c88836dcdce55b3f61d1804b1"}] }
```

7.57 istransferon

This call is used to check if currency transfer is allowed

```
{ "op": "istransferon", "params": [] }
```

7.58 isbuyon

This call is used to check if buying of currency is allowed

```
{ "op": "isbuyon", "params": [] }
```

7.59 issellon

This call is used to check if selling of currency is allowed

```
{ "op": "issellon", "params": [] }
```

7.60 isimporton

This call is used to check if currency import is allowed

```
{ "op": "isimporton", "params": [] }
```

7.61 isexporton

This call is used to check if currency export is on

```
{ "op": "isexporton", "params": [] }
```

7.62 hasextlink

This call is used to check if a specific address has and external link associated with it, the payload is as follows:

```
{ "op": "hasextlink", "params": [{"name": "toro",  
"value": "0xf3cdfc4a1dce2d98ff878971626b798279954c43"}, {"name": "crypto",  
"value": "0x34b92ef3509dcdac0e7fc58f247299ba67f3a22b"}] }
```

7.63 getnumberofextlink

This call is used to get the total number of external links associated with an address; the payload is as follows:


```
{ "op": "getnumberofextlink", "params": [{"name": "toro",  
"value": "0xf3cdfc4a1dce2d98ff878971626b798279954c43"}] }
```

7.64 getextlinkindex

This call is used to get the index of the external link of an address; the payload is as follows:

```
{ "op": "getextlinkindex", "params": [{"name": "toro",  
"value": "0xf3cdfc4a1dce2d98ff878971626b798279954c43"}, {"name": "crypto",  
"value": "0x34b92ef3509dcdac0e7fc58f247299ba67f3a22b"}] }
```

7.65 getextlinkbyindex

This call is used to get the external link of an address by its index, the payload is as follows:

```
{ "op": "getextlinkbyindex", "params": [{"name": "toro",  
"value": "0xf3cdfc4a1dce2d98ff878971626b798279954c43"}, {"name": "index", "value": 0}] }
```

7.66 iscryptoextlinked

This call is used to check if a specific cryptocurrency is externally linked, the payload is as follows:

```
{ "op": "iscryptoextlinked", "params": [{"name": "crypto",  
"value": "0x34b92ef3509dcdac0e7fc58f247299ba67f3a22b"}] }
```