
Wulet Documentation

Release 0.1.0

Valeriy Dubrava

Oct 26, 2018

Contents:

1	Common token	1
1.1	Tables	1
2	WU token	5
2.1	Actions	5
3	Loyalty token	7
3.1	Actions	7
4	Exchange	9
4.1	Tables	9
4.2	Actions	11
5	PUSH API	15
5.1	Add APNS token device	15
5.2	Delete APNS token device	15

1.1 Tables

1.1.1 Get balance

```
curl --request POST \  
  --url https://api-wulet.unblocking.io/v1/chain/get_table_rows \  
  --data '{"code": "TOKEN_ACCOUNT", "scope": "USER_ACCOUNT", "table": "accounts",  
↪ "json": true}'
```

Where:

- TOKEN_ACCOUNT — token account
- USER_ACCOUNT — user whose balance should be received

You will get response like:

```
{  
  "rows": [{  
    "balance": "100.000000 TEST",  
    "blocked": 20000000  
  }, {  
    "balance": "10.0000 FOO",  
    "blocked": 25000  
  }, {  
    "balance": "1.00 BAR",  
    "blocked": 75  
  }  
],  
  "more": false  
}
```

- balance field contains tokens that user have with symbol and precision

- blocked field contains blocked tokens without symbol and precision

To get available tokens that user can spend, blocked should be subtracted from balance. In the example above available user balances are 80.000000 TEST, 7.5000 FOO and 0.25 BAR.

1.1.2 Get token info

```
curl --request POST \
  --url https://api-wulet.unblocking.io/v1/chain/get_table_rows \
  --data '{"code": "TOKEN_ACCOUNT", "scope": "TOKEN_SYMBOL", "table": "stat", "json": true}'
```

Where:

- TOKEN_ACCOUNT — token account
- TOKEN_SYMBOL — symbol for which to get info

You will get response like:

```
{
  "rows": [{
    "supply": "30000.00 TEST",
    "max_supply": "100000.00 TEST",
    "issuer": "testowner",
    "info": {
      "name": "Symbol for API demonstration",
      "url": "https://wulet.readthedocs.io/en/develop/",
      "logo_url": "https://www.lenta.com/public/2015/img/icons/lenta_logo.png"
    }
  ]},
  "more": false
}
```

- supply field contains tokens that are already distributed
- max_supply field contains maximum number of tokens that can be distributed
- issuer field contains token owner that can distribute undistributed tokens
- info field contains token information
 - name is a full token name
 - url is a link to the store that distributes this token
 - logo_url is a link to the token logo

1.1.3 Get tokens list

```
curl --request POST \
  --url https://api-wulet.unblocking.io/v1/chain/get_table_rows \
  --data '{"code": "TOKEN_ACCOUNT", "scope": "TOKEN_ACCOUNT", "table": "symbols", "json": true}'
```

Where:

- TOKEN_ACCOUNT — token account

You will get response like:

```
{
  "rows": [{
    "symbol": "6,TEST"
  }, {
    "symbol": "4,FOO"
  }, {
    "symbol": "2,BAR"
  }
],
  "more": false
}
```

symbol field contains token name and its precision

2.1 Actions

2.1.1 Transfer

Transfer tokens from one account to another

```
{
  "code": "TOKEN_ACCOUNT",
  "action": "transfer",
  "args": {
    "from": "FROM_ACC",
    "to": "TO_ACC",
    "quantity": "100.0000 WU"
  }
}
```

Where:

- code - account where token code is deployed
- from - from account
- to - to account
- quantity - how many tokens to transfer

3.1 Actions

3.1.1 Burn

Destroys some amount of tokens

```
{
  "code": "TOKEN_ACCOUNT",
  "action": "transfer",
  "args": {
    "owner": "FROM_ACC",
    "value": "100.0000 TKN"
  }
}
```

Where:

- code - account where loyalty token code is deployed
- owner - account who wants to burn tokens
- value - how many tokens to burn

4.1 Tables

4.1.1 Pairs

To get a list of exchangeable pairs on the exchange, you need to get table:

```
POST /v1/chain/get_table_rows HTTP/1.0
Host: api-wuлет.unblocking.io

{
  "code": "exch.account",
  "scope": "exch.account",
  "table": "pairs",
  "lower_bound": "",
  "upper_bound": "",
  "limit": 10,
  "json": true
}
```

where:

- `code` — account of exchange,
- `scope` — should be same as exchange account,
- `table` — name of requested table, should be `pairs`,
- `lower_bound` — JSON representation of lower bound value of key, defaults to first,
- `upper_bound` — JSON representation of upper bound value value of key, defaults to last,
- `limit` — the maximum number of rows to return,
- `json` — returns JSON if `true` and binary if `false`,

You will get response like:

```
{
  "rows": [
    {
      "id": 0,
      "base_symbol": "4,WU",
      "quote_symbol": "4,A"
    },
    {
      "id": 1,
      "base_symbol": "4,A",
      "quote_symbol": "4,WU"
    }
  ],
  "more": false
}
```

where:

- rows — an array of pairs exchanged on the exchange,
- id — id of pair,
- base_symbol — precision and symbol name of base currency,
- quote_symbol — precision and symbol name of quote currency.

4.1.2 Markets

To get a list of offers on the exchange, filtered by pair, you need to:

```
POST /v1/chain/get_table_rows HTTP/1.0
Host: api-wulet.unblocking.io
```

```
{
  "code": "exch.account",
  "scope": 123,
  "table": "markets",
  "index_position": "2",
  "key_type": "float64",
  "lower_bound": "",
  "upper_bound": "",
  "limit": 10,
  "json": true
}
```

where:

- code — account of exchange,
- scope — id of exchanged pair from Pairs table,
- table — name of requested table, should be markets,
- index_position — index number to sort rows: 1 to sort by id, 2 to sort by price,
- key_type — key type of index: i64 for 1 index, float64 for 2 index,
- lower_bound — JSON representation of lower bound value of key, defaults to first,
- upper_bound — JSON representation of upper bound value value of key, defaults to last,

- `limit` — the maximum number of rows to return,
- `json` — returns JSON if `true` and binary if `false`,

You will get response like:

```
{
  "rows": [
    {
      "id": 0,
      "manager": "buyer1",
      "base": "2.0000 WU",
      "quote_symbol": "4,A",
      "price": "5.000000000000000000"
    },
    {
      "id": 1,
      "manager": "buyer1",
      "base": "3.0000 WU",
      "quote_symbol": "4,A",
      "price": "6.333333333333333333"
    },
    {
      "id": 2,
      "manager": "buyer1",
      "base": "1.0000 WU",
      "quote_symbol": "4,A",
      "price": "8.000000000000000000"
    }
  ],
  "more": false
}
```

where:

- `rows` — an array of trades on the exchange,
- `id` — id of trade,
- `manager` — account name of user who created the trade offer,
- `base` — base currency,
- `quote_symbol` — symbol of quote currency,
- `price` — price of the base currency against the quote,

4.2 Actions

4.2.1 Add trade

To add new trade to exchange you need to push action:

```
{
  "code": "exch.account",
  "action": "createx",
  "args": {
    "creator": "buyeraccount",
```

(continues on next page)

(continued from previous page)

```
"base_deposit": "10.1000 WU",
"quote_deposit": "50.3000 AIR"
}
}
```

where:

- code — account of exchange contract,
- action — performed action. should be `createx`,
- creator — your account,
- base_deposit — base currency in special format (as in example),
- quote_deposit — quote currency in same format,

4.2.2 Accept specified trade

To accept specified trade you need to push action:

```
{
  "code": "exch.account",
  "action": "spec.trade",
  "args": {
    "id": 123,
    "seller": "buyeraccount",
    "sell_symbol": "4,AIR",
    "receive": "10.1000 WU"
  }
}
```

where:

- code — account of exchange contract,
- action — performed action. should be `spec.trade`,
- id — id of trade,
- seller — your account,
- sell_symbol — precision and symbol name of quote currency in special format (as in example),
- receive — base currency in special format (as in example).

4.2.3 Market order trade

To get specified amount of tokens (market order) you need to push action:

```
{
  "code": "exch.account",
  "action": "market.trade",
  "args": {
    "seller": "buyeraccount",
    "sell_symbol": "4,AIR",
    "receive": "10.1000 WU"
  }
}
```

(continues on next page)

(continued from previous page)

```
}
}
```

where:

- `code` — account of exchange contract,
- `action` — performed action. should be `market.trade`,
- `seller` — your account,
- `sell_symbol` — precision and symbol name of quote currency in special format (as in example),
- `receive` — base currency you want to receive in special format (as in example).

4.2.4 Limit order trade

To get tokens for specified amount of another tokens (limit order) you need to push action:

```
{
  "code": "exch.account",
  "action": "limit.trade",
  "args": {
    "seller": "buyeraccount",
    "sell": "50.3000 AIR",
    "receive_symbol": "4,WU"
  }
}
```

where:

- `code` — account of exchange contract,
- `action` — performed action. should be `limit.trade`,
- `seller` — your account,
- `sell` — quote currency you want to receive in special format (as in example),
- `receive_symbol` — precision and symbol name of base currency in special format (as in example).

4.2.5 Cancel trade

To cancel your trade you need to push action:

```
{
  "code": "exch.account",
  "action": "cancelx",
  "args": {
    "id": 123,
    "base_symbol": "4,WU",
    "quote_symbol": "4,AIR"
  }
}
```

where:

- `code` — account of exchange contract,

- `action` — performed action. should be `cancelx`,
- `id` — id of the canceled trade,
- `base_symbol` — precision and symbol name of base currency in special format (as in example),
- `quote_symbol` — precision and symbol name of quote currency in special format (as in example).

5.1 Add APNS token device

```
POST /api/v1/user-devices HTTP/1.0
Host: api-wulet.unblocking.io

{
  "account": "iostestddg11",
  "token": "c6a0cef6e29d7b38040c843d2296b7ecdf3d66f5d1b62e3cffb64a2edadd6d05"
}
```

where:

- account - EOS account of user used at Wulet Platform,
- token - APNS device token of user.

5.2 Delete APNS token device

```
DELETE /api/v1/user-devices/TOKEN?account=ACCOUNT HTTP/1.0
Host: api-wulet.unblocking.io
X-Signature: SIGNATURE
```

where:

- TOKEN - EOS account of user used at Wulet Platform,
- ACCOUNT - APNS device token of user,
- SIGNATURE - APNS device token of user.